AD-A055 822 DUKE UNIV DURHAM N C ADAPTIVE SIGNAL DETECTION LAB F/G 9/3
A COMPARISON OF TWO NOISE CANCELING SYSTEM ALGORITHMS.(U)
DEC 77 R P MESSINA N00014-75-C-0191
UNCLASSIFIED TR-14 NL

1 OF 2
ADA
055822

School of
ENGINEERING
DUKE UNIVERSITY

TR-14

A COMPARISON OF TWO NOISE CANCELING SYSTEM ALGORITHMS

by

Rachel Patricia Messina
Department of Electrical Engineering

Prepared under:
Office of Naval Research (Code 222)
Contract No. N000-14-75-C-0191

Eruditio et Religio

DUKE UNIVERSITY

ADAPTIVE SIGNAL DETECTION LABORATORY

Department of Electrical Engineering
School of Engineering

Technical Report No. 14

A COMPARISON OF TWO NOISE CANCELING SYSTEM ALGORITHMS

by

Rachel Patricia Messina

December 1977

Approved:
L.W. Nolte
Principal Investigator

Prepared under: Office of Naval Research (Code 222)
Contract No. N00014-75-C-0191

This report was also a thesis submitted in partial fulfillment of the requirements for the Master of Science degree in the Department of Electrical Engineering in the Graduate School of Arts and Sciences of Duke University, 1977.

ABSTRACT

The purpose of this study is to compare two noise canceling system algorithms when the number of inputs to the system is small. The expected value algorithm is an estimate of the optimum solution for minimum mean square error analysis. The LMS (least mean square) algorithm was developed by Widrow (Widrow et al., 1975) to avoid the matrix inversion inherent in the optimum solution given by the Wiener-Hopf equation. The comparison between the two algorithms is measured by the rate of convergence of the mean square error and the minimum error after five hundred iterations. → next page

A two input noise canceling system is investigated in the first part of the study and a three input system is studied in the second part. A two input noise canceling system has one input that contains a mixture of the signal and noise and a second input which is noise alone. The noise only input is multiplied by a weight, W, and subtracted from the first input to produce an estimate of the signal. The three input system adds another noise only input and a corresponding weight.

The expected value and LMS algorithms attempt to determine the Wiener-Hopf weight vector through different methods. The expected value algorithm estimates the necessary expected values, substitutes them into the Wiener-Hopf equation, and performs the necessary matrix inversion. The LMS algorithm uses gradient techniques and the method of steepest descent to avoid the complexity of the matrix inversion.

A computer simulation of a dc signal in Gaussian noise provided the comparison between the two algorithms. The tradeoff in terms of performance for simplicity in the LMS algorithm was discovered to be great. The expected value algorithm consistently produced better results in terms of faster rates of convergence and smaller minimum errors after five hundred iterations.

The LMS algorithm presented problems in implementation because of two parameter values that had to be chosen. A value for the initial weights had to be chosen since the weights were determined using only previous data. This value was not as important when considering such a large number of iterations as was the value of μ, the term that controls the rate of convergence. This term must be within a specific range in order to guarantee convergence of the algorithm. This range is small when only considering five hundred iterations. Large errors produced by the algorithm were also a problem.

In conclusion, the expected value algorithm was judged to be the better of the two. The differences in performance were great especially when the signal-to-noise ratio or the correlation between the noises was small. If a "correct" value of μ is chosen, the results may be similar, but if not, the results can be extremely different from those of the expected value algorithm. Thus for a small number of inputs, where the matrix to be inverted is of small dimension, it is better in terms of performance to use the approximation to the optimum solution given by the expected value algorithm.

ACKNOWLEDGEMENTS

I would like to thank my advisor, Dr. L. W. Nolte, for his encouragement, guidance, and support of this research. I would like to thank also Dr. Jean Bullier and Dr. Sang C. Lee for their suggestions and guidance. The support of the Office of Naval Research is also acknowledged.

I wish to thank Mrs. Linda Hayes of Duke University for typing the final manuscript. Special thanks are also due to my fellow graduate students for their friendship, encouragement, and moral support.

R.P.M.

CONTENTS

APPENDICES

LIST OF FIGURES

LIST OF TABLES

A COMPARISON OF TWO NOISE CANCELING SYSTEM ALGORITHMS

CHAPTER I

INTRODUCTION

An important concept in signal detection theory is the noise canceling system which is used to eliminate noise from a contaminated signal. The noise canceling system consists of one input containing the contaminated signal and a varying number of noise alone inputs. Several algorithms have been developed in order to decrease the number of computations and thus increase the speed of convergence to the minimum error.

Early work in least squares estimation was done by Wiener in 1942 when he recognized that the solution of the Wiener-Hopf equation determined the optimum estimate for stochastic processes (Kailath, 1974). This was later extended to stationary and nonstationary processes. Kalman also contributed to the work by developing a filter which modeled the covariance of the signal process (Kailath, 1974). The LMS (least mean square) adaptive algorithm was developed in 1959 by Widrow and Hoff at Stanford University (Widrow _et_ _al_., 1975). Work on adaptive systems continued in the 1960's with major contributions by Lucky at Bell Laboratories (Widrow, _et_ _al_., 1975). This research compares the LMS algorithm with an estimate to the optimum solution for two and three input noise canceling systems.

A general n-input noise canceling system consists of one input with both signal and noise and (n-1) noise only inputs. Each noise only input is multiplied by a weight and subtracted from the first input in order to obtain an estimate of the signal. The system is shown in Figure 1.1.

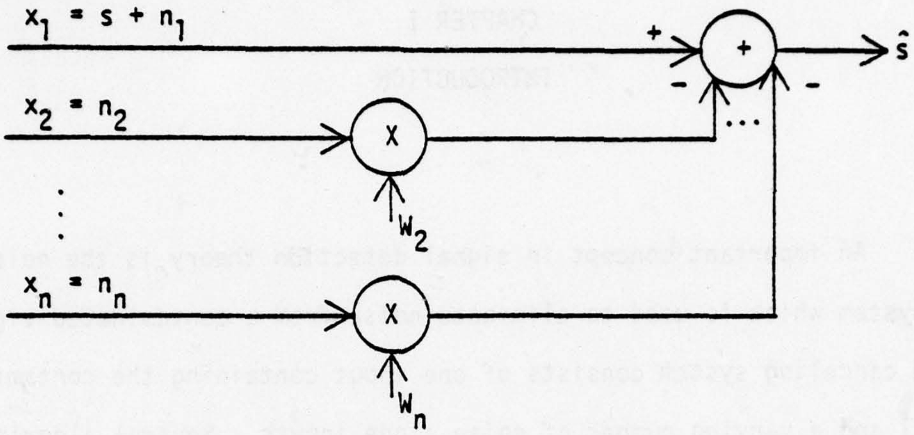


Figure 1.1. An n-input Noise Canceling System.

The weights are chosen so that the signal estimate, $\hat{s}$, is a linear least mean square estimate. That is, it is desirable to minimize

$$E[\varepsilon^2] = E[(\hat{s}-s)^2] \tag{1.1}$$

through a linear combination of the inputs. The optimum solution is given by the Wiener-Hopf equation:

$$\underline{W}^* = \underline{R}^{-1}\underline{P} \tag{1.2}$$

where $\underline{W}^*$ is the optimum weight vector:

$$\underline{W}^* = \begin{bmatrix} W_2^* \\ W_3^* \\ \cdot \\ \cdot \\ \cdot \\ W_n^* \end{bmatrix}. \tag{1.3}$$

$\underline{X}$ is the input vector:

$$\underline{X} = \begin{bmatrix} x_2 \\ x_3 \\ \vdots \\ x_n \end{bmatrix} \tag{1.4}$$

$\underline{R}$ is the input correlation matrix:

$$\underline{R} = E[\underline{X}\,\underline{X}^T] = \begin{bmatrix} E[x_2x_2] & \dots & E[x_2x_n] \\ E[x_3x_2] & & \vdots \\ \vdots & & \\ E[x_nx_2] & \dots & E[x_nx_n] \end{bmatrix} \tag{1.5}$$

and $\underline{P}$ is the cross correlation vector between the desired response, d and the input vector $\underline{X}$:

$$\underline{P} = E[d\underline{X}] = \begin{bmatrix} E[dx_2] \\ E[dx_3] \\ \vdots \\ E[dx_n] \end{bmatrix} \tag{1.6}$$

The minimum error can be found by substituting this value of $\underline{W}$ into the equation for the error:

$$\hat{s} = x_1 - \underline{W}^{*T}\,\underline{X} \tag{1.7}$$

and thus

$$\min E[(\hat{s}-s)^2] = E[(x_1 - \underline{W}^{*T}\,\underline{X}-s)^2] \tag{1.8}$$

The expected value and LMS (least mean square) algorithms attempt to determine the Wiener-Hopf weight vector through different methods. The expected value algorithm estimates the expected values of the $\underline{R}$ and $\underline{P}$ matrices and substitutes these into the Wiener-Hopf equation. Thus the weight vector is determined by actually taking the matrix inverse. The LMS algorithm discussed by Widrow et al., in "Adaptive Noise Cancelling: Principles and Applications" (Proceedings of the IEEE, December, 1975) uses gradient techniques and the method of steepest descent in order to avoid the complexity of the matrix inverse inherent in the optimum weight vector. The two algorithms are simulated on a PDP-11/45 computer at Duke University. A plot of the mean square error as a function of the number of iterations provides the main basis for the comparison.

Chapter II derives the algorithms for the two input noise canceling system. In this case, the R matrix is a 1x1 matrix. The optimum weight vector and the theoretical minimum mean square error are also derived. Chapter III presents the computer simulation and the results.

Chapter IV extends both algorithms to the three input system. The expected value algorithm is again the estimate to the optimum weight vector formed by estimating the necessary expected values. The LMS algorithm is Widrow's algorithm where the one element weight vector of the two input system has been replaced by a two element weight vector. Chapter V presents the results of the three input noise canceling system computer simulation.

The final chapter, Chapter VI, summarizes the results. Directions for possible future research are discussed.

CHAPTER II

DERIVATION OF THE ALGORITHMS

This research explores the problem of minimizing the mean square error of a noise canceling system. A two input noise canceling system is shown in Figure 2.1.

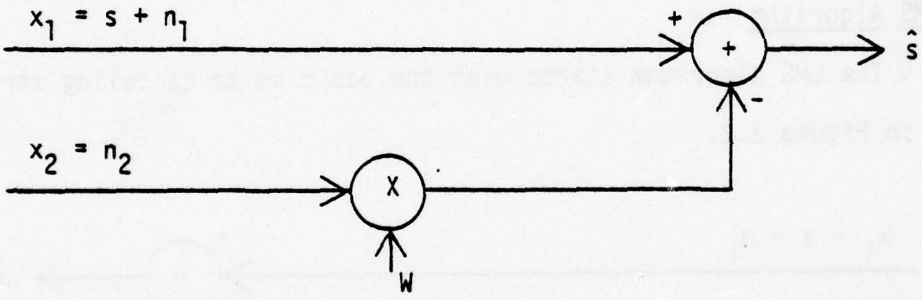


Figure 2.1. Two Input Noise Canceling System.

The signal, s, is uncorrelated with the noises, n_1 and n_2, which are correlated with each other. The weight, W, attempts to transform the noise n_2 so that it can be subtracted from the x_1 input to give an estimate of the signal. This estimate is defined as

$$\hat{s} = x_1 - Wx_2 \tag{2.1}$$

Thus the error is given

$$\varepsilon = \hat{s} - s \tag{2.2}$$

The mean square error is defined as the expected value of the squared error:

$$\varepsilon_{MS} = E[\varepsilon^2] = E[(\hat{s} - s)^2] \tag{2.3}$$

It is desired to derive the weight, W, in such a way that the estimate of the signal, $\hat{s}$, produced is a linear least mean square estimate. Thus the estimate attempts to minimize the mean square error $E[\varepsilon^2]$ through a linear combination of the inputs. The optimum solution is given by the Wiener-Hopf equation. This problem was studied by Widrow and the LMS (least mean square) algorithm was developed. This algorithm is used in many noise canceling systems and was one of the two studied in this research.

1. LMS Algorithm

The LMS algorithm starts with the basic noise canceling structure shown in Figure 2.2.

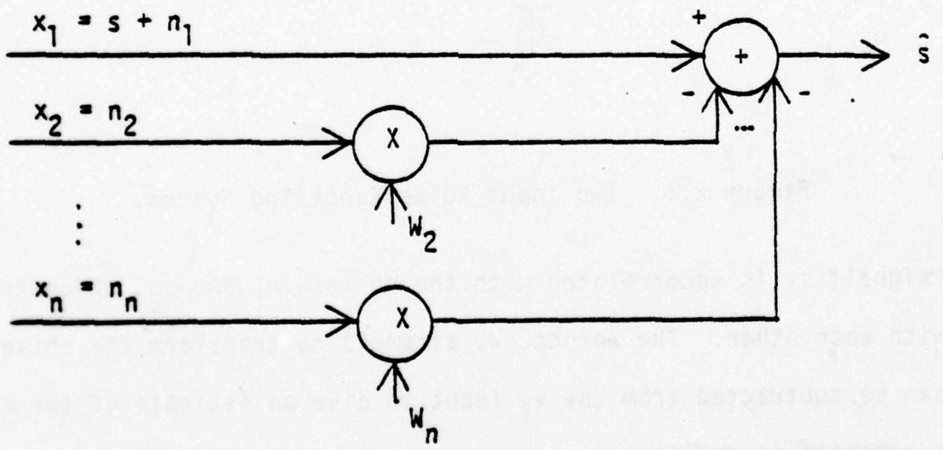


Figure 2.2. Noise Canceling Structure for LMS Algorithm

The LMS algorithm as described in Widrow's "Adaptive Noise Cancelling: Principles and Applications," attempts to adjust the weight vector, $\underline{W}$, to minimize the mean square error. The method of steepest descent is used as a practical way to find close approximate solutions to the Wiener-Hopf equation:

$$\underline{W}^* = \underline{R}^{-1}\underline{P} \tag{2.4}$$

where $\underline{W}^*$ is the optimum weight vector, $\underline{R}$ is the input correlation matrix

$$\underline{R} = E[\underline{X}_j\underline{X}_j{}^T] \tag{2.5}$$

and $\underline{P}$ is the cross correlation vector between the desired response and the input, or $\underline{X}_j$, vector

$$\underline{P} = E[d_j\underline{X}_j] \tag{2.6}$$

as defined in Chapter I for the jth inputs.

In order to avoid the matrix inversion required to find the optimum weight vector, the LMS algorithm sets the next weight vector, $\underline{W}_{j+1}$, equal to the present weight vector $\underline{W}_j$ plus another term proportional to the negative gradient, ∇_j, of the weight vector. Thus

$$\underline{W}_{j+1} = \underline{W}_j - \mu\underline{\nabla}_j \tag{2.7}$$

where μ is a parameter that controls stability and the rate of convergence. (For a study of convergence to the optimum solution see Appendix A). The instantaneous gradient $\underline{\nabla}_j$ is also estimated by assuming that the squared error $\varepsilon_j{}^2$ is an approximation of the mean square error $E[\varepsilon_j{}^2]$. Thus the LMS algorithm uses an estimate $\hat{\underline{\nabla}}_j$ of the true gradient.

From Figure 2.2, the estimated signal, $\hat{s}$, for an n input system is given by

$$\hat{s} = x_1 - \underline{W}^T\underline{X} \tag{2.8}$$

Substituting (2.8) into the value of the error, ε, given by (2.2) yields

$$\varepsilon = n_1 - \underline{W}^T\underline{X} \tag{2.9}$$

Taking the partial derivative of the error with respect to $\underline{W}$ produces

$$\frac{\partial\varepsilon}{\partial\underline{W}} = - \underline{X} \tag{2.10}$$

The partial derivative of the squared error with respect to $\underline{W}$ is given by

$$\frac{\partial\varepsilon^2}{\partial\underline{W}} = 2\varepsilon\,\frac{\partial\varepsilon}{\partial\underline{W}} = -\,2\varepsilon\underline{X} \tag{2.11}$$

The estimated gradient at the jth iteration, $\hat{\underline{\nabla}}_j$, is set equal to the partial derivative of the squared error with respect to $\underline{W}$. Thus

$$\hat{\underline{\nabla}}_j = \frac{\partial\varepsilon_j^{\,2}}{\partial\underline{W}} = -2\varepsilon_j\underline{X}_j \tag{2.12}$$

Finally from the method of steepest descent, the weight vector can be determined:

$$\begin{aligned} \underline{W}_{j+1} &= \underline{W}_j - \mu\hat{\underline{\nabla}}_j \\ \underline{W}_{j+1} &= \underline{W}_j + 2\mu\varepsilon_j\underline{X}_j \end{aligned} \tag{2.13}$$

When the noise canceling system has only two inputs as shown in Figure 2.3, the expression for the weight vector reduces to a scalar quantity:

$$W_{2_{j+1}} = W_{2_j} + 2\mu\varepsilon_j x_{2_j} \tag{2.14}$$

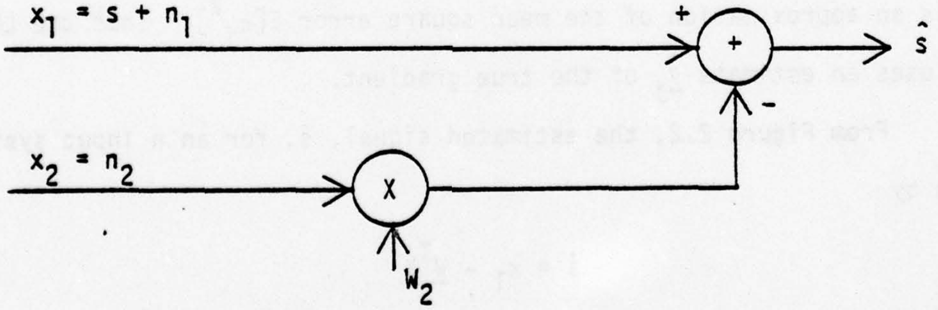


Figure 2.3. Two Input Noise Canceling System.

This noise canceling system can also be written in terms of a feedback system. A general two input feedback noise canceling system is shown in Figure 2.4. The input d_j is the desired response and ε_j is the error.

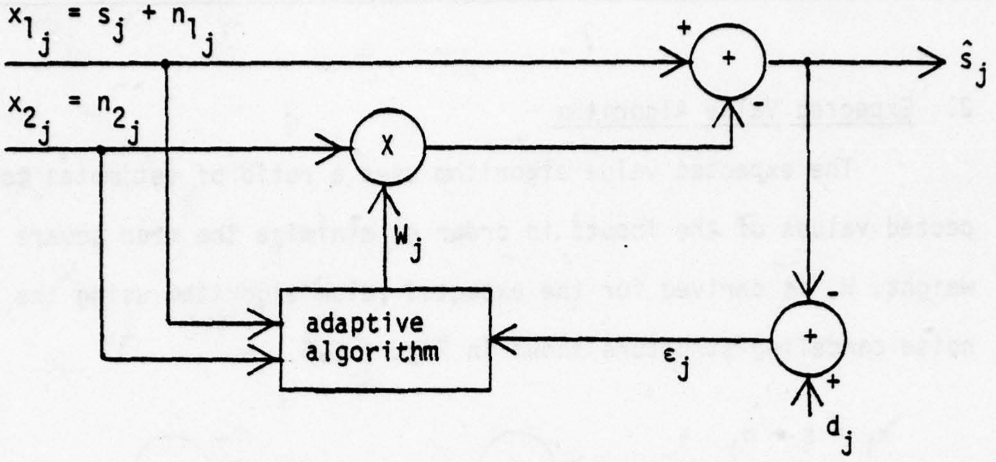


Figure 2.4. General Two Input Feedback Noise Canceling System.

Since the desired input is actually the unknown signal, d_j is often set equal to x_1 (Widrow <u>et</u> <u>al</u>., 1975). When this is done, the system is not a feedback system in the true sense because the error term is now given by

$$\begin{aligned} \varepsilon_j &= d_j - \hat{s}_{LMS_j} \\ \varepsilon_j &= x_{1_j} - \hat{s}_{LMS_j} \\ \varepsilon_j &= W_{LMS_j} x_{2_j} \end{aligned} \tag{2.15}$$

Thus the weight, W_{j+1}, is given in terms of inputs only when (2.14) is used to define the weight.

Since the noise canceling system employing the LMS algorithm as defined by (2.14), is not an actual feedback system, another algorithm was developed. This algorithm, the expected value algorithm, also attempts to minimize the mean square error. However, a feedback approach was not taken. Instead two weights and a brute force method of solving the matrix inverse avoided by the LMS algorithm were used. For a two input system this matrix is one dimensional so the inverse is quite simple.

2. Expected Value Algorithm

The expected value algorithm uses a ratio of estimates to the expected values of the inputs in order to minimize the mean square error. The weight, W, is derived for the expected value algorithm using the two input noise canceling structure shown in Figure 2.5.

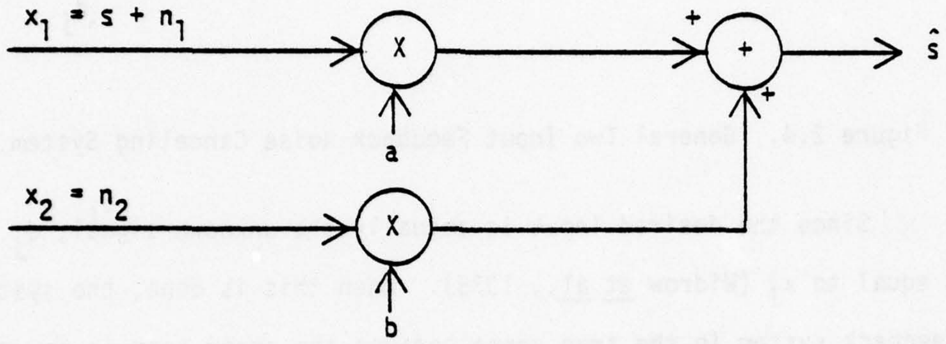


Figure 2.5. Two Input Noise Canceling System.

For this system, the estimated signal, $\hat{s}$, is given by

$$\hat{s} = ax_1 + bx_2 \tag{2.16}$$

The orthogonality principle gives

$$E[(\hat{s} - s)x_1] = 0 \tag{2.17}$$

and

$$E[(\hat{s} - s)x_2] = 0 \tag{2.18}$$

Substituting the value of $\hat{s}$ given by (2.16) into (2.17) and (2.18) produces

$$aE[x_1x_1] + bE[x_2x_1] - E[sx_1] = 0 \tag{2.19}$$

and

$$aE[x_1x_2] + bE[x_2x_2] - E[sx_2] = 0 \tag{2.20}$$

Writing (2.19) and (2.20) in matrix form gives

$$\begin{bmatrix} E[x_1x_1] & E[x_2x_1] \\ E[x_1x_2] & E[x_2x_2] \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} E[sx_1] \\ E[sx_2] \end{bmatrix} \tag{2.21}$$

Taking the matrix inverse, equation (2.21) can be solved for a and b. Thus

$$a = \frac{E[sx_1]E[x_2^2] - E[x_1x_2]E[sx_2]}{E[x_1^2]E[x_2^2] - [E[x_1x_2]]^2} \tag{2.22}$$

and

$$b = \frac{E[x_1^2]E[sx_2] - E[x_1x_2]E[sx_1]}{E[x_1^2]E[x_2^2] - [E[x_1x_2]]^2} \tag{2.23}$$

By hypothesis, $E[sn_1] = 0$ and $E[sn_2] = 0$. Thus $E[sx_2] = 0$ and

$$E[sx_1] = E[s(s + n_1)] = E[s^2] + E[sn_1] = E[s^2] \tag{2.24}$$

Substituting these values into (2.22) and (2.23) gives

$$a = \frac{E[s^2]E[x_2^2]}{E[x_1^2]E[x_2^2] - [E[x_1x_2]]^2} \tag{2.25}$$

and

$$b = -\frac{E[s^2]E[x_1x_2]}{E[x_1^2]E[x_2^2] - [E[x_1x_2]]^2} \tag{2.26}$$

Let

$$\alpha = \frac{E[s^2]}{E[x_1^2]E[x_2^2] - [E[x_1x_2]]^2} \tag{2.27}$$

Substituting the value of α into (2.25) and (2.26) gives

$$a = \alpha E[x_2^2] \tag{2.28}$$

and

$$b = -\alpha E[x_1x_2] \tag{2.29}$$

Looking again at the noise canceling system diagram and replacing the values of a and b as determined in (2.28) and (2.29) yields

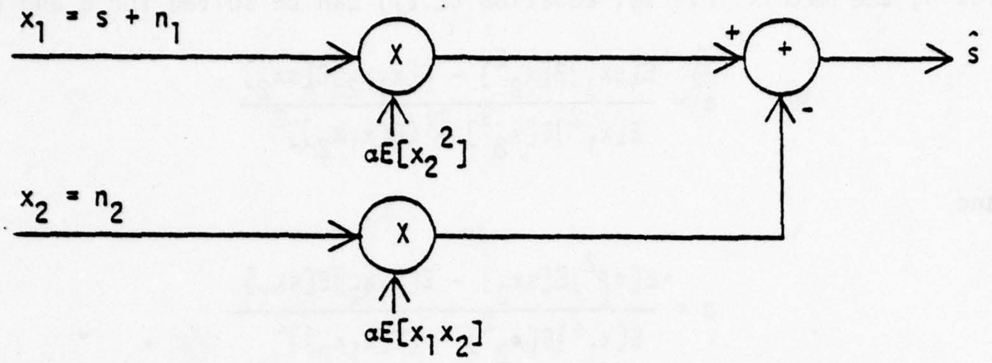


Figure 2.6. Two Input Noise Canceling System for the Expected Value Algorithm.

or

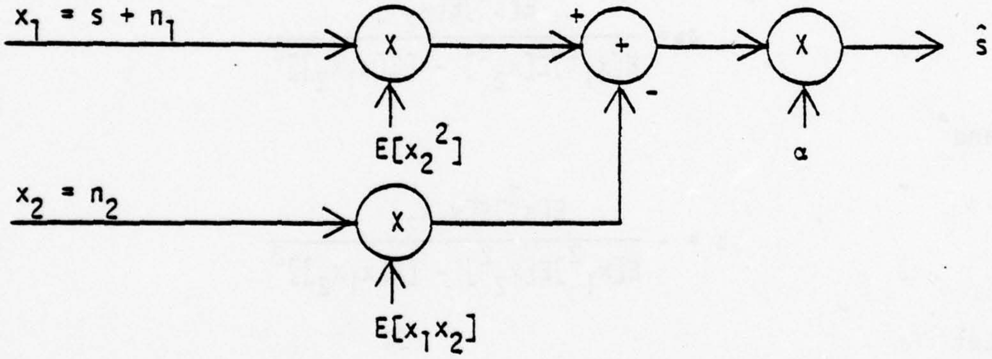


Figure 2.7. Two Input Noise Canceling System for the Expected Value Algorithm.

The term α can be simplified by recognizing several relationships. One input to the noise canceling system is x_1 where

$$x_1 = s + n_1 \tag{2.30}$$

Squaring both sides of (2.30) gives

$$x_1^2 = s^2 + n_1^2 + 2sn_1 \tag{2.31}$$

Since $E[sn_1] = 0$, taking the expected value of both sides of (2.31) gives

$$E[x_1^{\,2}] = E[s^2] + E[n_1^{\,2}] \tag{2.32}$$

The second input to the noise canceling system is x_2 where

$$x_2 = n_2 \tag{2.33}$$

Since $x_2^{\,2} = n_2^{\,2}$, taking the expected value of both sides gives $E[x_2^{\,2}] = E[n_2^{\,2}]$. Also since $x_1x_2 = sn_2 + n_1n_2$, taking the expected value yields $E[x_1x_2] = E[n_1n_2]$.

The correlation between n_1 and n_2 is defined as ρ_{12} where

$$\rho_{12} = \frac{E[n_1n_2]}{\sqrt{E[n_1^{\,2}]E[n_2^{\,2}]}} \tag{2.34}$$

Squaring and rearranging (2.34) produces

$$[E[n_1n_2]]^2 = \rho_{12}^{\,2}E[n_1^{\,2}]E[n_2^{\,2}] \tag{2.35}$$

Substituting (2.35) into (2.27) gives

$$\alpha = \frac{E[s^2]}{E[n_2^{\,2}]} \qquad \frac{1}{[E[s^2] + E[n_1^{\,2}]] - \rho_{12}^{\,2}E[n_1^{\,2}]} \tag{2.36}$$

Since $E[x_2^{\,2}] = E[n_2^{\,2}]$, let

$$\beta = \alpha E[x_2^{\,2}] = \alpha E[n_2^{\,2}]$$

Thus

$$\beta = \frac{E[s^2]}{E[s^2] + E[n_1^{\,2}](1 - \rho_{12}^{\,2})} \tag{2.37}$$

The system of Figure 2.7 can be reduced again to produce a multiplier of one in the top branch as shown in Figure 2.8.

The term β is found to depend on the apriori knowledge in terms of the signal-to-noise ratio and the correlation between the two noise terms. Dividing the numerator and denominator of (2.37) by $E[n_1^{\,2}]$ and recognizing

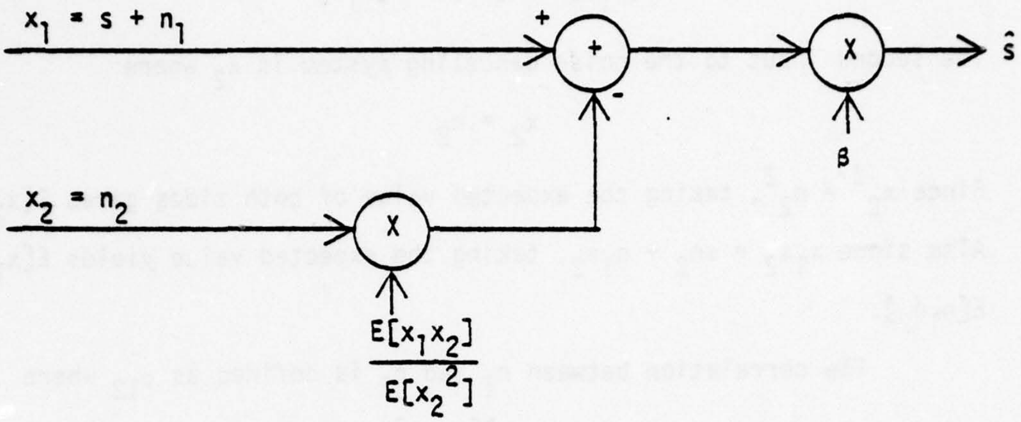


Figure 2.8. Two Input Noise Canceling System for the Expected Value Algorithm.

that the signal-to-noise ratio (S/N) is the term $E[s^2]/E[n_1^2]$ yields

$$\beta = \frac{\frac{S}{N}}{\frac{S}{N} + 1 - \rho_{12}^{2}} \tag{2.38}$$

Thus the weight, W, for the expected value algorithm is given by

$$W = \frac{E[x_1x_2]}{E[x_2^2]} \tag{2.39}$$

where the term β must also be a multiplier of the output of the system.

In summary, the expected value algorithm uses a ratio of the expected values of the inputs in order to determine the weight, W. However, since expected values are mathematical concepts, they are estimated for the computer simulation in terms of the jth sequence of inputs. The two input noise canceling system with the expected value algorithm is shown in Figure 2.9.

Both the LMS and expected value algorithms have only one weight which is multiplied by the noise only input. Therefore an interesting problem is to find the optimum weight for a linear least mean square estimate.

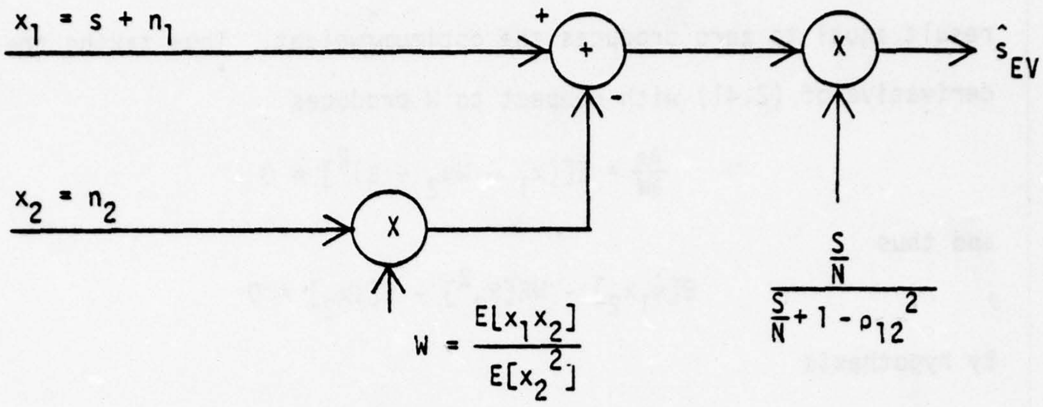

$x_1 = s + n_1$
+
+
-
X
$\hat{s}_{EV}$
$x_2 = n_2$
X
$W = \frac{E[x_1 x_2]}{E[x_2^2]}$
$\frac{\frac{S}{N}}{\frac{S}{N} + 1 - \rho_{12}^2}$

Figure 2.9. Expected Value Algorithm.

3. Optimum Weight Vector

The noise canceling structure is shown in Figure 2.10.

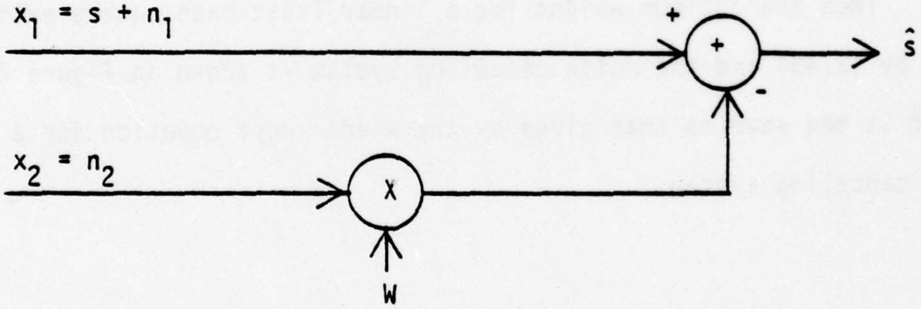


Figure 2.10. Two Input Noise Canceling System.

The error, ε, for a linear least mean square estimate is given by

$$\varepsilon = E[(\hat{s} - s)^2] \qquad (2.40)$$

where $\hat{s}$ is defined in (2.1). Substituting the value of $\hat{s}$ given by (2.1) into (2.40) yields

$$\varepsilon = E[(x_1 - Wx_2 - s)^2] \qquad (2.41)$$

Taking the derivative of the error with respect to the weight and setting the result equal to zero produces the optimum weight. Thus taking the partial derivative of (2.41) with respect to W produces

$$\frac{\partial \varepsilon}{\partial W} = E[(x_1 - Wx_2 - s)^2] = 0$$

and thus

$$E[x_1x_2] - WE[x_2{}^2] - E[sx_2] = 0 \qquad (2.42)$$

By hypothesis

$$E[sn_1] = E[sn_2] = E[sx_2] = 0$$

Solving for W in (2.42) yields

$$W_{opt} = \frac{E[x_1x_2]}{E[x_2{}^2]} \qquad (2.43)$$

Thus the optimum weight for a linear least mean square estimate is given by (2.43) and the noise canceling system is shown in Figure 2.11. This weight is the same as that given by the Wiener-Hopf equation for a two input noise canceling system.

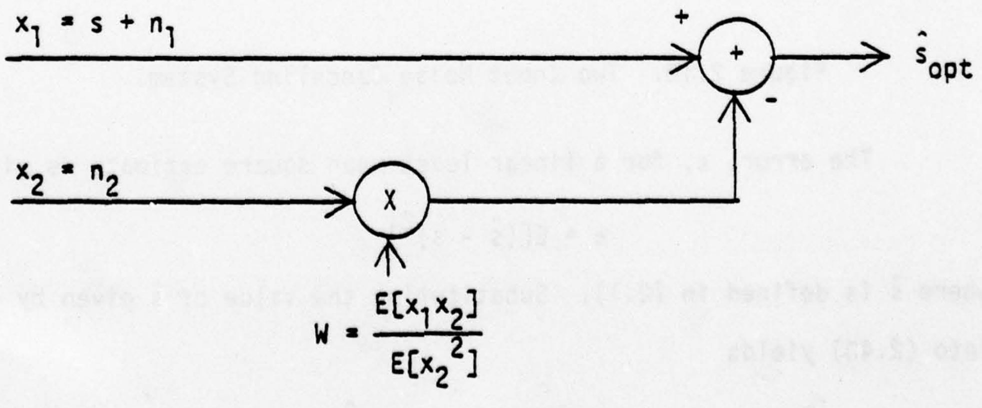


Figure 2.11. Optimum Weight Vector for the Two Input Noise Canceling System.

Notice that the optimum weight and the theoretical weight for the expected value algorithm are the same. The expected value algorithm, however, has a multiplier of β which is not present in the noise canceling structure of Figure 2.11. When β = 1, the expected value algorithm would be the same as the optimum if the expected values could be computed directly. The weight for the LMS algorithm will converge to the optimum weight as the number of iterations increase (see Appendix A).

Since both the LMS and expected value algorithms have weights which converge to the same value, the theoretical minimum mean square error can be found.

4. Theoretical Minimum Mean Square Error

The theoretical minimum mean square error for both algorithms is the same (since the weights converge to the value given by the optimum weight) and can be determined from the optimum weight vector. Using the noise canceling structure of Figure 2.10 and the optimum weight vector given by (2.43), the minimum mean square error, ϵ_{MMS}, can be computed:

$$\epsilon_{MMS} = E[(\hat{s} - s)^2]$$

$$\epsilon_{MMS} = E[(x_1 - \frac{E[x_1x_2]}{E[x_2^2]} x_2 - s)^2]$$

and

$$\epsilon_{MMS} = E[(n_1 - \frac{E[n_1n_2]}{E[n_2^2]} n_2)^2] \quad (2.44)$$

If

$$\gamma = \frac{E[n_1n_2]}{E[n_2^2]} \quad (2.45)$$

then substituting (2.45) into (2.44) produces

$$\epsilon_{MMS} = E[(n_1 - \gamma n_2)^2] \tag{2.46}$$

Expanding (2.46) yields

$$\epsilon_{MMS} = E[n_1{}^2] - 2\gamma E[n_1 n_2] + \gamma^2 E[n_2{}^2] \tag{2.47}$$

The correlation coefficient, ρ_{12}, is defined in (2.34) and can be rewritten in terms of $E[n_1 n_2]$. That is:

$$E[n_1 n_2] = \rho_{12}\sqrt{E[n_1{}^2]E[n_2{}^2]} \tag{2.48}$$

Substituting (2.48) into (2.45) yields

$$\gamma = \rho_{12}\sqrt{\frac{E[n_1{}^2]}{E[n_2{}^2]}} \tag{2.49}$$

Using (2.48) and (2.49) the minimum error can be simplified:

$$\epsilon_{MMS} = E[n_1{}^2] - 2\left[\rho_{12}\sqrt{\frac{E[n_1{}^2]}{E[n_2{}^2]}}\right]\rho_{12}\sqrt{E[n_1{}^2]E[n_2{}^2]} + \left[\rho_{12}{}^2\,\frac{E[n_1{}^2]}{E[n_2{}^2]}\right]E[n_2{}^2]$$

and thus

$$\epsilon_{MMS} = E[n_1{}^2](1 - \rho_{12}{}^2) \tag{2.50}$$

The theoretical minimum mean square error for the noise canceling system of Figure 2.10 is given by (2.50). The minimum error depends only on the power of the noise mixed with the signal and the correlation between the two noises. As the correlation decreases to zero, the error approaches its largest value. When the correlation actually equals zero, this is the worst case in terms of error. Thus the error would be the same if no noise canceling system had been used. When the correlation equals one, the error equals zero and the best case is realized. Thus as the correlation increases and the noise power decreases (implying increasing signal-to-noise ratio), the theoretical minimum mean square error also decreases.

For the expected value algorithm, Figure 2.10 assumes that $\beta = 1$. This is a reasonable assumption because β depends on apriori knowledge of the signal-to-noise ratio and the correlation coefficient ρ_{12}. If this knowledge is known, a better estimate can be obtained.

Using the noise canceling structure of Figure 2.9 and the value of β from (2.38), the theoretical minimum mean square error for $\beta \neq 1$ can be computed. From Figure 2.9, the estimated signal, $\hat{s}$, is given by:

$$\hat{s} = \beta[x_1 - \frac{E[x_1x_2]}{E[x_2^2]} x_2] \qquad (2.51)$$

Substituting (2.51) into (2.40) gives

$$\varepsilon_{MMS} = E[(\beta x_1 - \frac{E[x_1x_2]}{E[x_2^2]} \beta x_2 - s)^2] \qquad (2.52)$$

Using (2.43) and (2.48), equation (2.52) can be simplified:

$$\varepsilon_{MMS} = E[n_1^2](1 - \rho_{12}^2)\beta^2 + E[s^2](\beta - 1)^2 \qquad (2.53)$$

If $\beta = 1$, then (2.53) reduces to (2.50).

5. Summary of the Algorithms

The algorithms were derived in a theoretical manner, but are summarized here in the forms which were implemented for the computer simulation. The inputs, weights, and estimates are given in terms of the jth iteration.

A. LMS Algorithm

The weight vector for the LMS algorithm was defined in (2.13) in terms of the error:

$$\underline{W}_{j+1} = \underline{W}_j + 2\mu\varepsilon_j\underline{X}_j \qquad (2.13)$$

However, since the signal, s, is not generally known, it is desirable to replace the error ε_j in (2.13) by $\hat{s}_j$. Then the weight vectors can be shown to converge to the same value. Thus for the jth iteration

$$\epsilon_j = \hat{s}_j - s_j \tag{2.54}$$

Replacing ϵ_j by $\hat{s}_j$ in (2.13) gives

$$\underline{W}_{\hat{s}_{j+1}} = \underline{W}_{\hat{s}_j} + 2\mu\epsilon_j\underline{X}_j + 2\mu s_j\underline{X}_j \tag{2.55}$$

and in general

$$\underline{W}_{\hat{s}_j} = \underline{W}_0 + 2\mu\sum_{i=0}^{j-1}\epsilon_i\underline{X}_i + 2\mu\sum_{i=0}^{j-1}\underline{X}_i s_i \tag{2.56}$$

Taking the expected value of both sides of (2.56) produces

$$E[\underline{W}_{\hat{s}_j}] = \underline{W}_0 + 2\mu\sum_{i=0}^{j-1}E[\epsilon_i\underline{X}_i] + 2\mu\sum_{i=0}^{j-1}E[\underline{X}_i s_i] \tag{2.57}$$

By hypothesis $E[\underline{X}\ s] = 0$. Therefore

$$E[\underline{W}_{\hat{s}_j}] = \underline{W}_0 + 2\mu\sum_{i=0}^{j-1}E[\epsilon_i\underline{X}_i] \tag{2.58}$$

Therefore replacing ϵ_j by either $\hat{s}_j$ or $\epsilon_j = \hat{s}_j - s_j$ produces the same weight solution.

The weight vector used throughout the remainder of this paper replaces ϵ_j by $\hat{s}_j$ and for the two input system is given by

$$W_{j+1} = W_j + 2\mu\hat{s}_j x_{2_j} \tag{2.59}$$

The weight vector can also be written in terms of the data inputs alone by substituting for the value of $\hat{s}_j$ from (2.1). Thus

$$W_{j+1} = W_j(1 - 2\mu x_{2_j}{}^2) + 2\mu x_{1_j} x_{2_j} \tag{2.60}$$

The weight vector, W_{j+1}, is computed during the computer simulation after determining $\hat{s}_{LMS}$. This value is stored in a vector so that it can be used to compute $\hat{s}_{LMS}$ for the next iteration. Figure 2.12 shows the LMS algorithm for the jth iteration of the computer simulation.

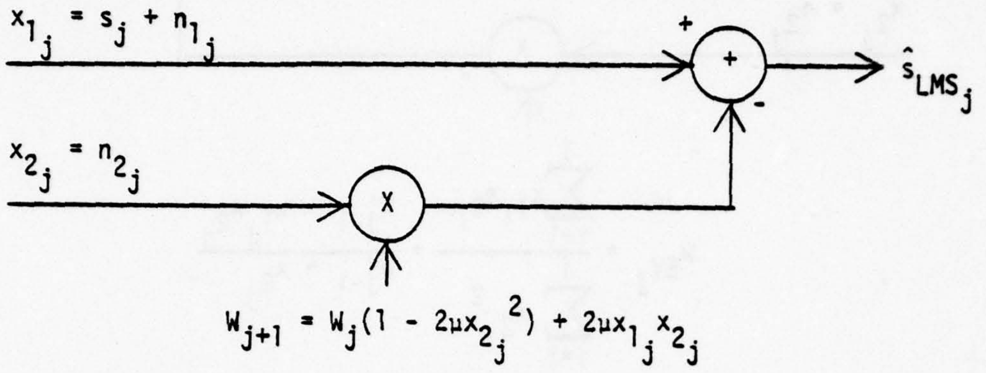

$x_{1_j} = s_j + n_{1_j}$
$x_{2_j} = n_{2_j}$
+
+
-
X
$\hat{s}_{LMS_j}$
$W_{j+1} = W_j(1 - 2\mu x_{2_j}{}^2) + 2\mu x_{1_j} x_{2_j}$

Figure 2.12. LMS Algorithm as Computer Simulated.

B. Expected Value Algorithm

When the expected value algorithm was implemented on the PDP-11/45 computer, the expected values were estimated using an iterative approximation. For example:

$$E[x_{1_j} x_{2_j}] = \frac{\sum_{i=1}^{j} x_{1_i} x_{2_i}}{j} \tag{2.61}$$

The term β was set equal to one for reasons explained later. Thus for the jth iteration, the expected value algorithm is implemented as shown in Figure 2.13.

The expected value algorithm can be written in a form which reveals the way in which the weight is updated:

$$W_{EV_j} = \frac{\sum_{i=1}^{j} x_{1_i} x_{2_i}}{\sum_{i=1}^{j} x_{2_i}{}^2} = \frac{W'_{j-1} + x_{1_j} x_{2_j}}{W''_{j-1} + x_{2_j}{}^2} = \frac{W_j{}'}{W_j{}''} \tag{2.62}$$

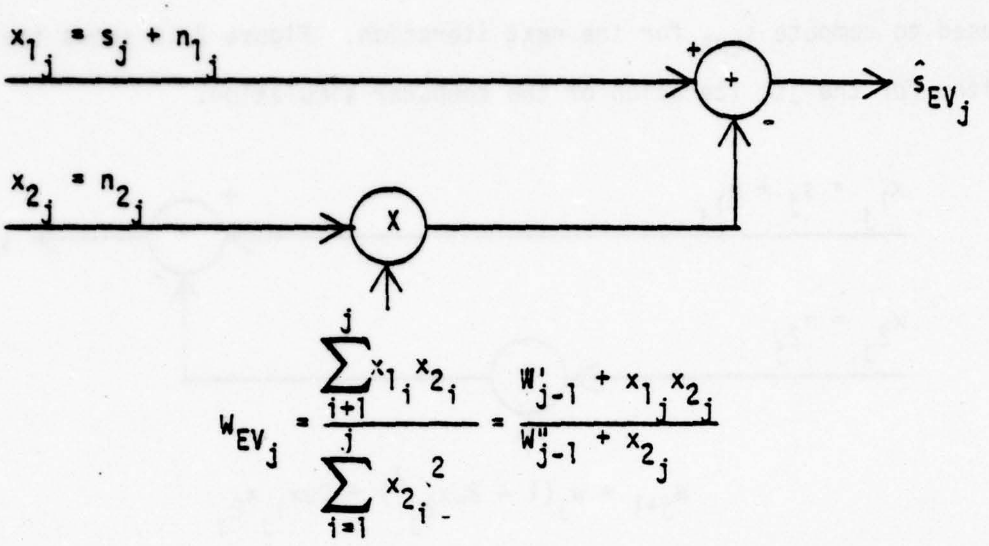

$x_{1_j} = s_j + n_{1_j}$
+
+
-
$\hat{s}_{EV_j}$
$x_{2_j} = n_{2_j}$
X
$W_{EV_j} = \frac{\sum_{i+1}^{j} x_{1_i} x_{2_i}}{\sum_{i=1}^{j} x_{2_i}^{2}} = \frac{W'_{j-1} + x_{1_j} x_{2_j}}{W''_{j-1} + x_{2_j}}$

Figure 2.13. Expected Value Algorithm as Computer Simulated.

This form also indicates the estimation of the expected values needed for the optimum weight vector defined in (2.43).

C. Comparison of the Algorithms

The optimum weight vector as given in Figure 2.11 will produce the best results in terms of mean square error estimation for the two input system. However, since the expected values are not known quantities, they must be estimated. The difference in performance between the expected value and LMS algorithms can be traced to the weights. The expected value algorithm makes two estimates (one for the numerator and another for the denominator) before dividing the two to obtain an estimate of the weight:

$$W_{EV_j} = \frac{\sum_{i=1}^{j} x_{1_i} x_{2_i}}{\sum_{i=1}^{j} x_{2_i}^{2}} = \frac{W'_{j-1} + x_{1_j} x_{2_j}}{W''_{j-1} + x_{2_j}^{2}} \tag{2.63}$$

The jth inputs are used to update the (j-1)st estimates before the division is made.

The LMS algorithm however, makes one estimate for the weight:

$$W_{LMS_j} = W_{LMS_{j-1}} + 2\mu\hat{s}_{j-1}x_{2_{j-1}} \tag{2.64}$$

or in terms of the inputs:

$$W_{LMS_j} = W_{LMS_{j-1}}(1 - 2\mu x_{2_{j-1}}{}^2) + 2\mu x_{1_{j-1}}x_{2_{j-1}} \tag{2.65}$$

Although the same terms, x_1x_2 and $x_2{}^2$, are being calculated, the two estimate approach of the expected value algorithm generally will produce a better estimate in fewer iterations. The expected value algorithm also uses the jth inputs to compute the jth weight, but the LMS algorithm only uses the data through the (j-1)st inputs. This is the reason the LMS algorithm can be written as a feedback system and the expected value algorithm can not.

CHAPTER III

COMPUTER SIMULATION OF TWO INPUT NOISE CANCELING SYSTEM

The two algorithms described and developed in Chapter II were simulated on a PDP-11/45 computer at Duke University. In order to make the algorithms more comparable, the noise canceling structure of Figure 2.10 was used for the simulation. The multiplier of β for the expected value algorithm, as shown in Figure 2.8, was set equal to one. If β is actually calculated, a better estimate could be obtained. However, for purposes of comparison it was desirable to use the same amount of apriori knowledge for both algorithms. The comparison was based on a graph with the mean square error plotted as a function of the number of iterations.

1. Problem Description

Values for the inputs to the noise canceling system of Figure 2.10 had to be chosen before beginning the computer simulation. A dc signal was used as the signal component of the system. The value of the signal was constant and arbitrarily chosen to be 10.0. The choice of a dc signal enabled a more comprehensive study since a frequency dependent signal, such as a sine wave, would have had the effects of changes in phase or frequency embedded in the results.

The noise was generated using the computer's random number generator. Uniformly distributed random numbers were obtained and then transformed to zero mean Gaussian random numbers by employing the Box-Muller equations. Each pair of Gaussian random numbers had a given variance and correlation and served as the noise components needed for the inputs to the noise canceling system.

The algorithms were implemented on the computer using several iterative estimates. The expected values needed for the expected value algorithm were computed using the approximation

$$E[x_{1_j}x_{2_j}] = \frac{\sum_{i=1}^{j} x_{1_i}x_{2_i}}{j} \tag{3.1}$$

The mean square error was also approximated:

$$E[(\hat{s}_j - s_j)^2] = \frac{\sum_{i=1}^{j} (\hat{s}_i - s_i)^2}{j} \tag{3.2}$$

The initial weight vector, W(1), for the LMS algorithm was set to a constant value of 5.0. This value was an arbitrary choice. Since the mean square error was computed for five hundred iterations, the effect of the initial weight vector washed out. If it was necessary to use a smaller number of iterations, the effect of the initial weight vector would be more important and thus it should be carefully chosen.

The LMS algorithm also required values for the parameter μ which controls the rate of convergence of the algorithm. In order for the two input system LMS algorithm to converge, the value of μ must be within a given range (see Appendix A) defined as

$$|1 - 2\mu E[x_2^2]| < 1 \tag{3.3}$$

As the value of μ decreases, the rate of convergence also decreases since

$|1 - 2\mu E[x_2^2]|$ approaches 1. Three values of μ were chosen for the computer simulation. These were values of .01, .001, and .0001 which correspond to graphs b, c, and d, respectively on each page of figures. For convenience in the computer simulation

$$E[n_1^2] = E[n_2^2] \tag{3.4}$$

Therefore when the signal-to-noise ratio is 10.0, all three μ values are within the range for convergence of the algorithm. When the signal-to-noise ratio is 1.0, μ values of .001 and .0001 are within the range, but a μ value of .01 results in equality in equation (3.3). When the signal-to-noise ratio is .1, only the μ value of .0001 is within the range of convergence. Equality in equation (3.3) results for a μ value of .001 and a μ value of .01 is outside the range for convergence of the algorithm.

The theoretical minimum mean square error is defined in (2.50) and depends only on the values of $E[n_1^2]$ and ρ_{12}. Thus when $\rho_{12} = 0$, the worst error is produced. Figure 3.1 shows the worst case for the signal-to-noise ratio of 10.0. When $\rho_{12} = 1$, the best case is realized and the noise is completely canceled. Therefore when ρ_{12} is between 0 and 1 in value, some noise canceling is achieved, but some error also exists.

With the noise canceling system described, the method of comparison was a graph of the mean square error as a function of the number of iterations. Each curve consisted of five hundred iterations. Twenty-five curves were averaged point by point before this ensemble average was plotted. In this way, the effect of a single ensemble (or particular sequence of random numbers) was less and the averaged curve was a smoother curve. The curves can be compared directly or the difference from the theoretical minimum mean square error after five hundred iterations can be used as a basis of comparison.

2. Performance Comparison

The graphs of the mean square error as a function of the number of iterations follow. They are arranged by pages with the same signal-to-noise ratio (S/N) and noise correlation (ρ_{12}) on each page. Each page contains a graph using the expected value algorithm and three graphs of the LMS algorithm, each of which employs a different μ value. The pages are arranged in order of decreasing signal-to-noise ratio and decreasing correlation coefficients. All of the graphs were plotted using the same scale in order to compare them directly. When the mean square error was too large to be within the bounds set by the scale, the curve was plotted as a straight line until the iteration when the value came within bounds. This does not mean that the mean square error was a constant value through this iteration, only that it was too large for the desired scale (see Appendix C). Table 3.1 gives representative values of the curves.

The difference between the minimum mean square errors of the curves is explained by the different rates of convergence. The theoretical minimum error is the same for each set of values of signal-to-noise ratio and noise correlation. Changes in the amount of noise correlation have little effect on the rate of convergence itself, but do change the value of the theoretical minimum error.

Figure 3.1 demonstrates the worst case ($\rho_{12} = 0.0$) for a signal-to-noise ratio of 10.0. The expected value algorithm shows rapid convergence to the theoretical minimum error of 10.0. After five hundred iterations, the actual error (with a value of 11.22) is very close to the theoretical. This is the worst case for this signal-to-noise ratio. Any larger value of noise correlation will improve the ability of the system to eliminate the noise. The LMS algorithm does not perform as well as the expected value algorithm because the rate of convergence is slower and the minimum error after five hundred iterations is larger. As the value of μ is decreased, the rate of convergence becomes less. However, the μ value of .001 produces a smaller error than does the μ value of .01. Although the rate of convergence is less, the minimum error is smaller due to fewer fluctuations in the weight vector.

Figures 3.2 through 3.4 all have the signal-to-noise ratio of 10.0. Each figure has a different value of noise correlation. For each figure, the expected value algorithm exhibits the best performance in terms of a lower minimum value and a faster rate of convergence. In each case the LMS algorithm with μ equal to .01 compares more favorably with the expected value algorithm in terms of rate of convergence. However, with μ equal to .001, the LMS algorithm compares better in terms of minimum error. The μ equal to .0001 case of the LMS algorithm has a slow rate of convergence and large initial and final errors. Thus this case is not as important in the comparison between the two algorithms.

When the signal-to-noise ratio is 10.0, small changes in correlation produce only small changes in the curves. As the correlation is reduced, the rate of convergence of all of the curves decreases. The minimum error after five hundred iterations becomes larger. The maximum error for the expected value algorithm never exceeded 100, but the LMS algorithm often produced large errors. As the correlation decreased, these maximum errors increased.

When the signal-to-noise ratio was lowered to 1.0, the comparison is more dramatic. Figures 3.5 through 3.7 display these results. Again the expected value algorithm produces the best results. The change in signal-to-noise ratio has scarcely affected the performance of this algorithm. However, the LMS algorithm has been affected by the change to a great degree. The μ equal to .01 case of the LMS algorithm produces a curve that is entirely off of the scale. This is due to the fact that the value of μ is outside of the range of convergence. The μ equal to .0001 curve has a very slow rate of convergence and the initial values are so large that the straight line effect is observed. Thus the μ equal to .001 curve is the only curve for the LMS algorithm that can be effectively compared with expected value algorithm. The rate of convergence is much slower than for the expected value algorithm and the minimum error is much larger. The LMS algorithm again produces large initial errors.

With a smaller signal-to-noise ratio, changes in noise correlations are more important. Small differences in correlation have a greater effect on the rate of convergence and the maximum and minimum errors than when the signal-to-noise ratio is larger. If the correlation is constant and the signal-to-noise ratio is lowered or if the signal-to-noise ratio is constant and the correlation decreases, the rate of convergence is slower and the maximum and minimum errors for five hundred iterations increase. This was true for all of the cases tested except that the maximum error for the expected value algorithm was never greater than 100.

When the signal-to-noise ratio was lowered to .10, the expected value algorithm alone produced curves with values small enough to be within the desired scale. The μ values of .01 and .001 are too large for the algorithm to converge. However, as the μ value gets smaller, the rate of convergence becomes too slow for convergence within five hundred iterations. Thus the μ value of .0001 also produces a curve with values too large for the scale. Small changes in correlation produce noticeable changes in the expected value algorithm curves. When the signal-to-noise ratio was 10.0, a change in correlation from .99 to .98 produced barely distinguishable changes in the curves. Nevertheless, the expected value algorithm produces good results in terms of a fast rate of convergence and a minimum error that approaches the theoretical minimum.

3. Summary of Two Input Results

In all of the cases tested, the expected value algorithm produced the superior results. The expected value algorithm consistently had a faster rate of convergence and a smaller minimum value. It also had two fewer inputs than the LMS algorithm since an initial weight vector, W(1), and a μ value did not have to be chosen. The expected value algorithm never produced error values greater than 100, but the LMS algorithm produced many values that were too large to be graphed on the desired scale.

The complexity of the two algorithms is approximately the same. In order to calculate the weight vector, each algorithm requires two multiplications. The expected value algorithm also required an additional division. However, in terms of performance, the extra division time is well worth the small trade off in speed. Each algorithm computes the same terms, x_1x_2 and x_2^2, but the expected value algorithm updates two estimates at each iteration before dividing the two to obtain an estimate of the weight, W. The LMS algorithm, however, only computes one estimate of the weight. Although the one estimate takes less time, the two estimate approach produces better performance.

The problem in choosing a μ value proved to be great. In general, as the μ value decreased, the rate of convergence also decreased, but the minimum values of the curves approached more closely the theoretical minimum error due to fewer fluctuations in the weight vector. Since only the first 500 iterations were used, the range of μ was even smaller than equation (3.3) would indicate. This small range proved to be an important drawback to the LMS algorithm. The large error values produced by some values of μ created storage and graphing problems that were not present in the expected value algorithm. For signal-to-noise ratios of 1.0 and .10, the reason that the larger values of μ generated curves off the scale was due to the fact that the μ value was not in the range for convergence. However, for small values of μ, the large errors were probably due to the slow rate of convergence of the weights.

Although the choice of the initial weight vector was not important in this study because of such a large number of iterations, it could have an effect when considering a smaller number. In any case, the effect of the choice is not as important as that of the μ value because it tends to wash out after several iterations. This is another value, however, which must be chosen for the LMS algorithm and which is not necessary for the expected value algorithm.

The comparison between the two algorithms is dramatic when the signal-to-noise ratio is high, but becomes even more so as the signal-to-noise ratio decreases. When the signal-to-noise ratio is 10.0, the results are closer if a correct μ value is chosen than when the signal-to-noise ratio is .10 and only the expected value algorithm can be said to converge within five hundred iterations. The performance for all signal-to-noise ratios and correlation values tested reveal that the expected value algorithm works better for a two input noise canceling system with a dc signal and Gaussian noise than does the LMS algorithm.

Table 3.1. Representative Values of the Curves Obtained: Two Input Noise Canceling System

Each page contains four graphs labeled a, b, c, and d. These have characteristic values given by several parameters:

$\frac{S}{N}$ - the signal-to-noise ratio

ρ_{12} - the correlation between the noises n_1 and n_2

TMIN - the theoretical minimum mean square error

Algorithm - the algorithm employed-either expected value or LMS

μ - the value of the parameter μ for the LMS algorithm

MAX - the maximum value of the curve

MIN - the minimum value of the curve

Figure	$\frac{S}{N}$	ρ_{12}	TMIN	Algorithm	μ	MAX	MIN
3.1a	10	0.0	10.0	expected value		100	11.22
b				LMS	.01	230.4	26.82
c				LMS	.001	242.0	23.68
d				LMS	.0001	259.1	117.4
3.2a	10	.95	.975	expected value		100	2.408
b				LMS	.01	142.6	16.04
c				LMS	.001	153.8	10.26
d				LMS	.0001	165.0	71.72
3.3a	10	.75	4.375	expected value		100	5.728
b				LMS	.01	162.3	19.99
c				LMS	.001	172.3	14.50
d				LMS	.0001	184.6	82.11
3.4a	10	.50	7.5	expected value		100	8.778
b				LMS	.01	185.7	23.68
c				LMS	.001	195.5	18.75
d				LMS	.0001	209.3	94.55
3.5a	1.0	.95	9.75	expected value		100	10.97
b				LMS	.01	3.019E21	1426
c				LMS	.001	1426	36.0
d				LMS	.0001	1536	95.03
3.6a	1.0	.85	27.75	expected value		100	28.52
b				LMS	.01	3.438E21	1526
c				LMS	.001	1526	56.88
d				LMS	.0001	1628	117.2

Table 3.1 cont'd.

Figure	$\frac{S}{N}$	ρ_{12}	TMIN	Algorithm	μ	MAX	MIN
3.7a	1.0	.75	43.75	expected value		100	44.06
b				LMS	.01	3.715E21	1623
c				LMS	.001	1623	75.52
d				LMS	.0001	1720	137.5
3.8a	.10	.99	19.9	expected value		100	20.88
b				LMS	.01	overflow	overflow
c				LMS	.001	3.277E21	13830
d				LMS	.0001	13830	146.0
3.9a	.10	.98	39.6	expected value		100	40.03
b				LMS	.01	overflow	overflow
c				LMS	.001	3.648E21	13940
d				LMS	.0001	13940	168.8
3.10a	.10	.97	59.1	expected value		100	58.71
b				LMS	.01	overflow	overflow
c				LMS	.001	3.993E21	14050
d				LMS	.0001	14050	191.3

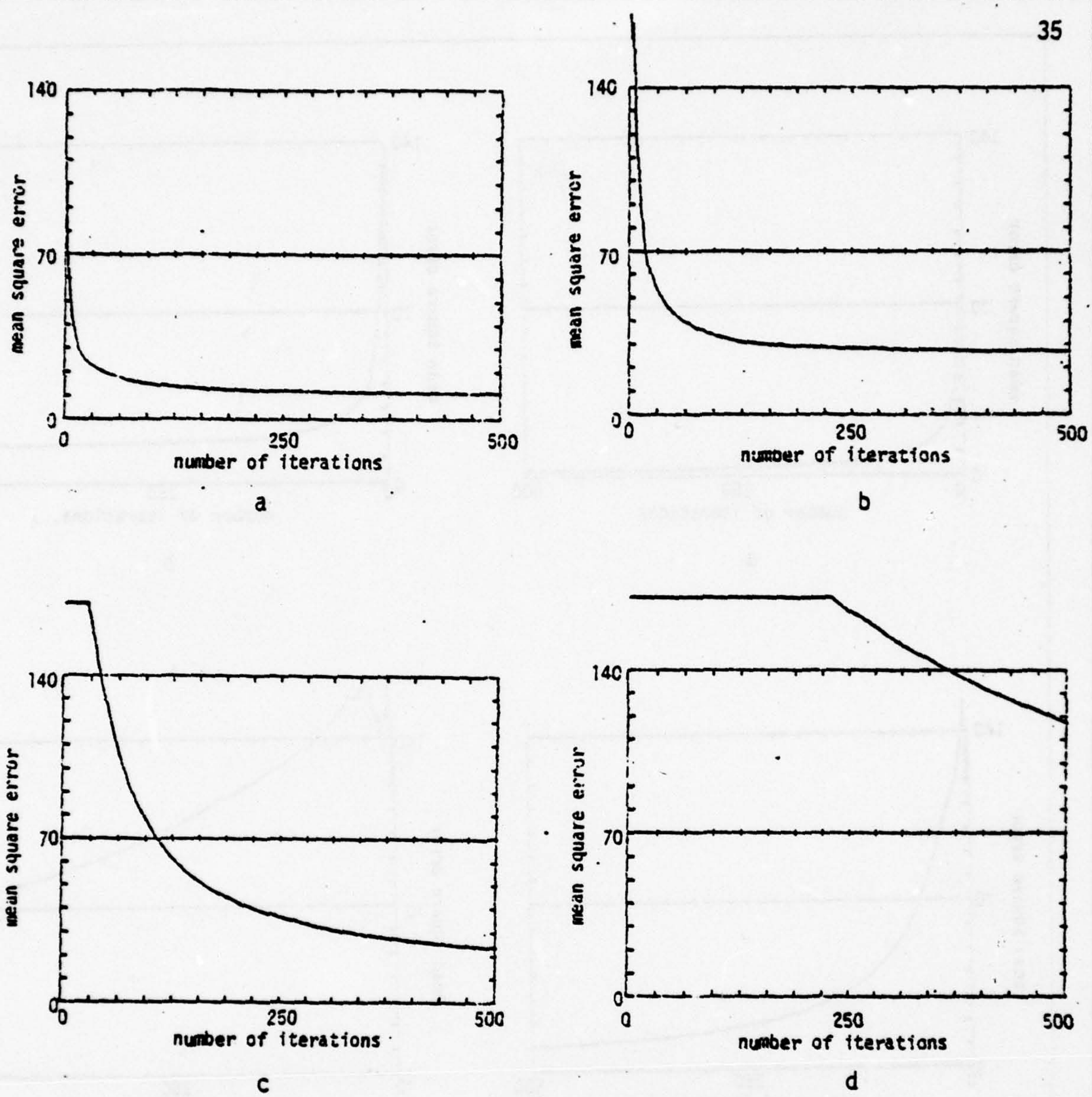


Figure 3.1. Computer Simulation of the Two Input System.

$S/N = 10.0$ and $\rho_{12} = 0.0$.

(a) Expected value algorithm.
(b) LMS algorithm, $\mu = .01$.
(c) LMS algorithm, $\mu = .001$.
(d) LMS algorithm, $\mu = .0001$.

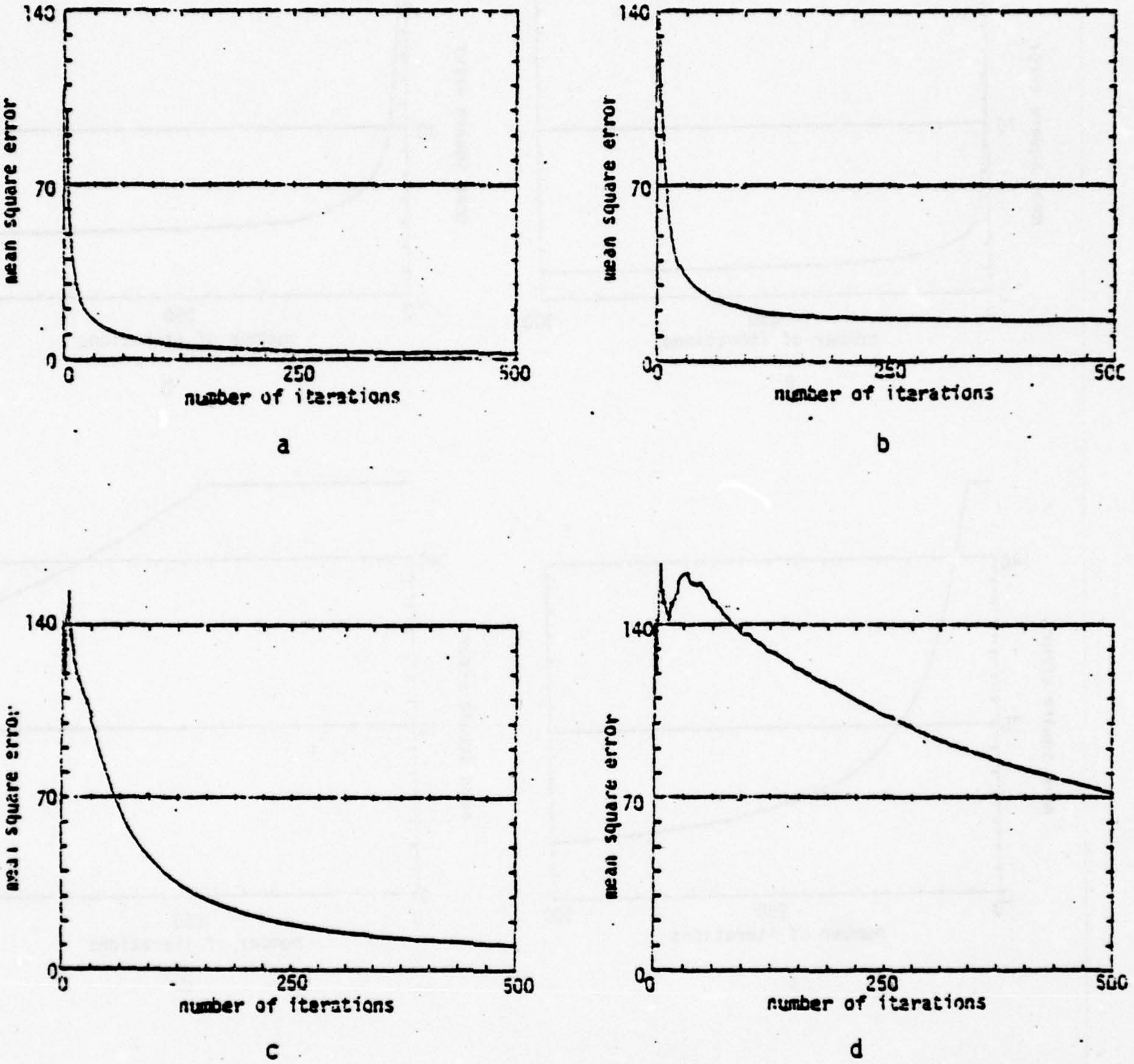


Figure 3.2. Computer Simulation of the Two Input System.
$S/N = 10.0$ and $\rho_{12} = .95$
(a) Expected value algorithm.
(b) LMS algorithm, $\mu = .01$.
(c) LMS algorithm, $\mu = .001$.
(d) LMS algorithm, $\mu = .00001$.

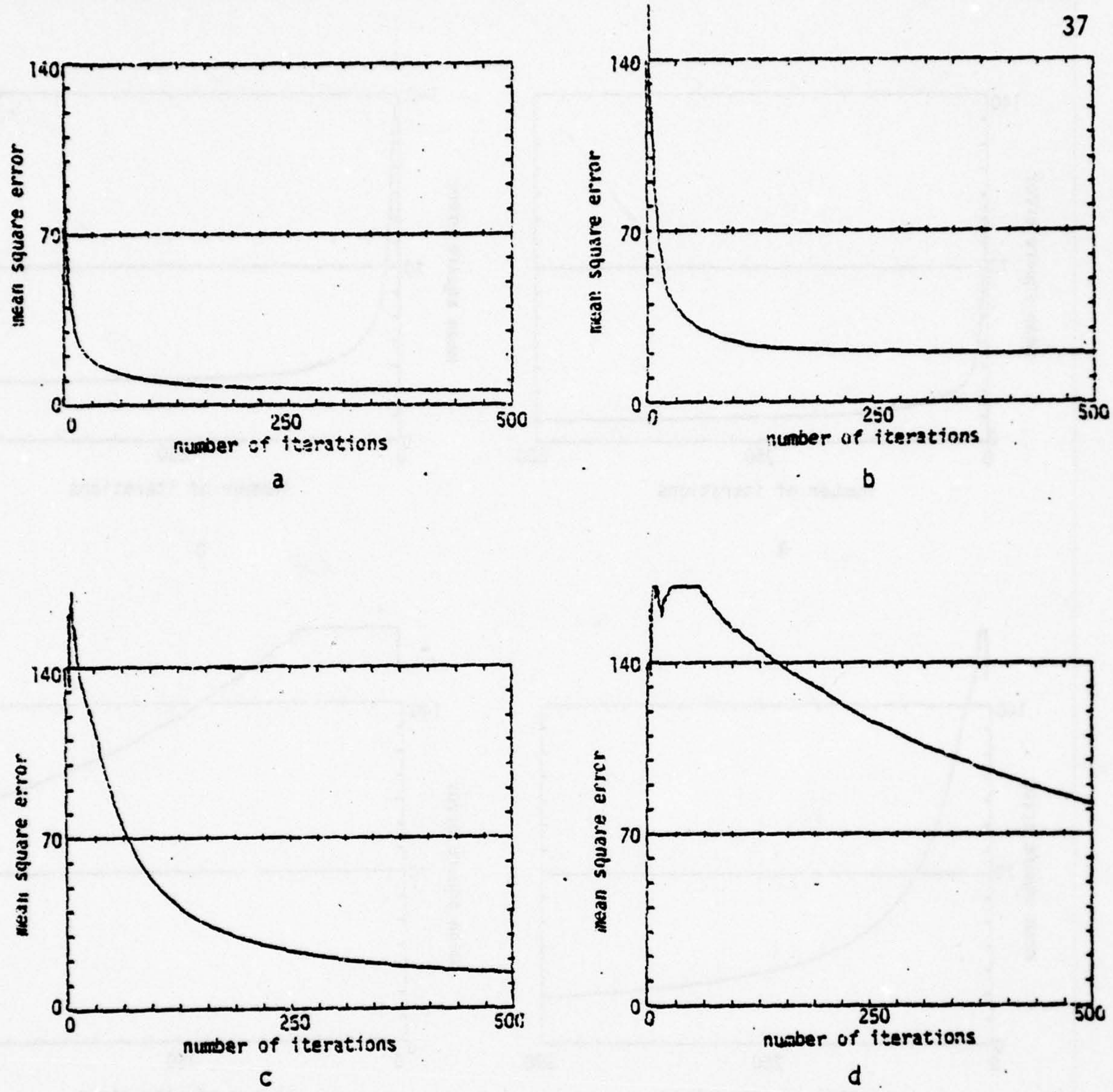


Figure 3.3. Computer Simulation of the Two Input System.

S/N = 10.0 and ρ_{12} = .75.

(a) Expected value algorithm.
(b) LMS algorithm, μ = .01.
(c) LMS algorithm, μ = .001.
(d) LMS algorithm, μ = .00001.

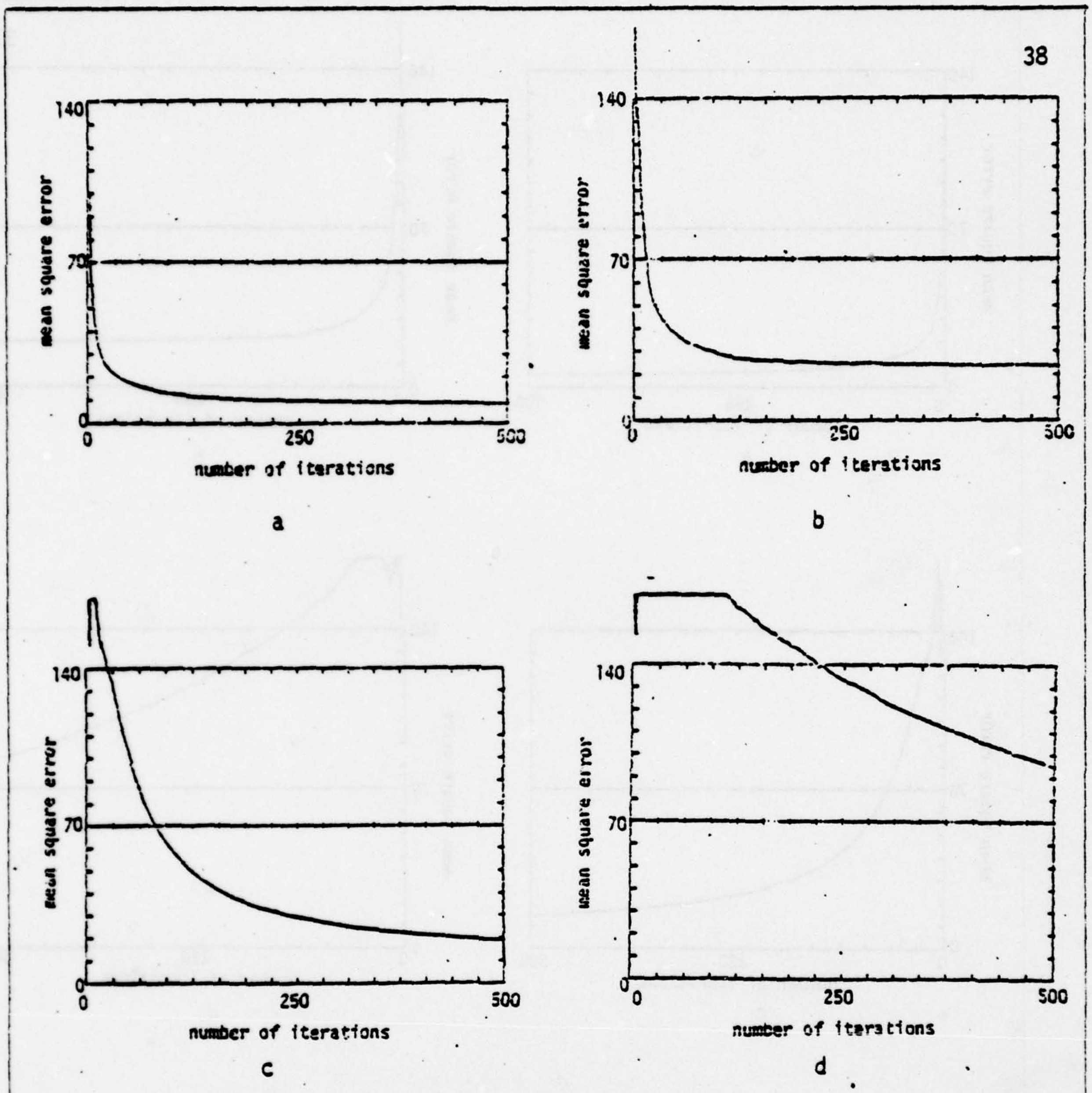


Figure 3.4. Computer Simulation of the Two Input System.
$S/N = 10.0$ and $\rho_{12} = .50$.
(a) Expected value algorithm.
(b) LMS algorithm, $\mu = .01$.
(c) LMS algorithm, $\mu = .001$.
(d) LMS algorithm, $\mu = .0001$.

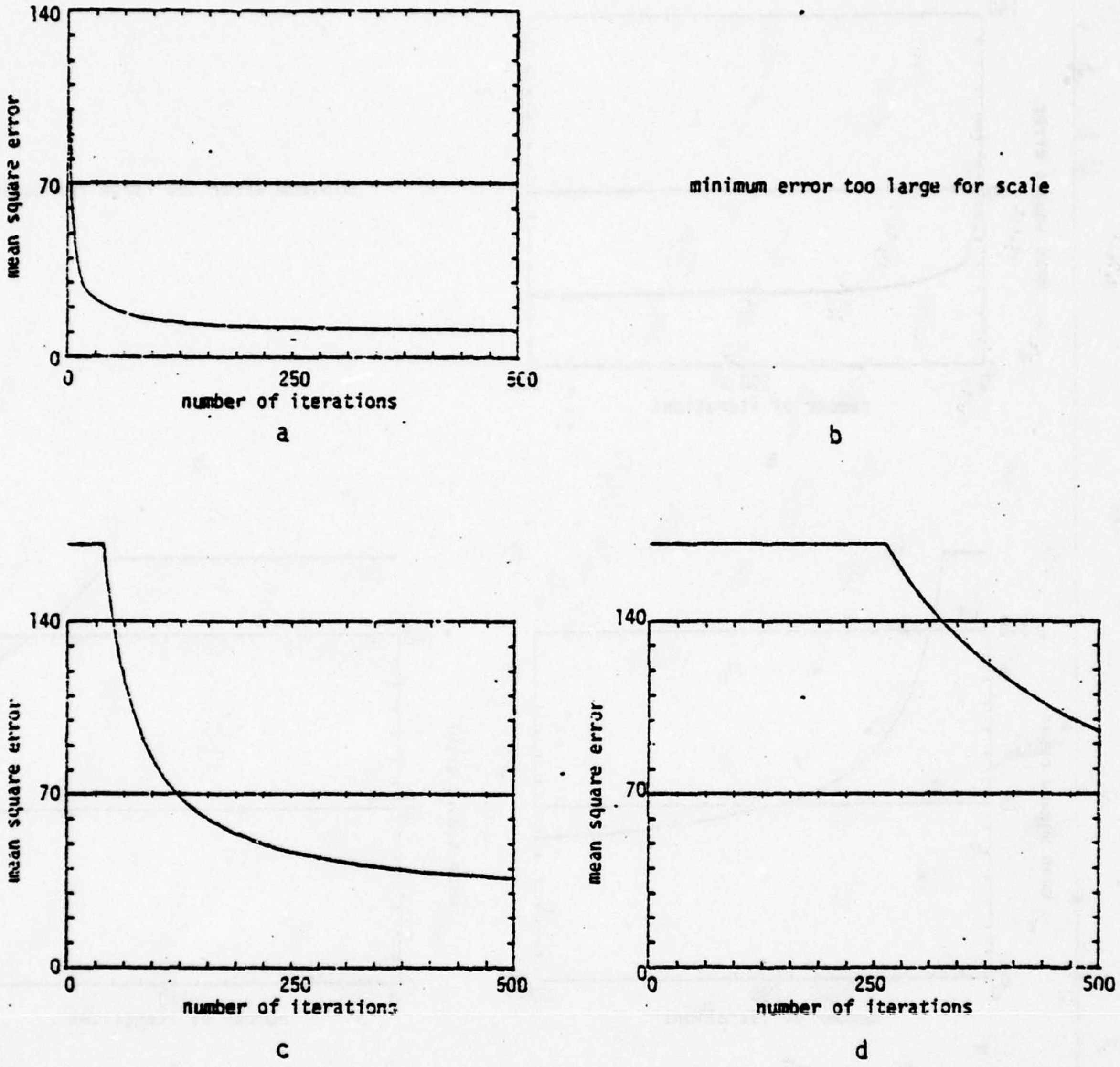


Figure 3.5. Computer Simulation of the Two Input System.
$S/N = 1.0$ and $\rho_{12} = .95$.

(a) Expected value algorithm.
(b) LMS algorithm, $\mu = .01$.
(c) LMS algorithm, $\mu = .001$.
(d) LMS algorithm, $\mu = .0001$.

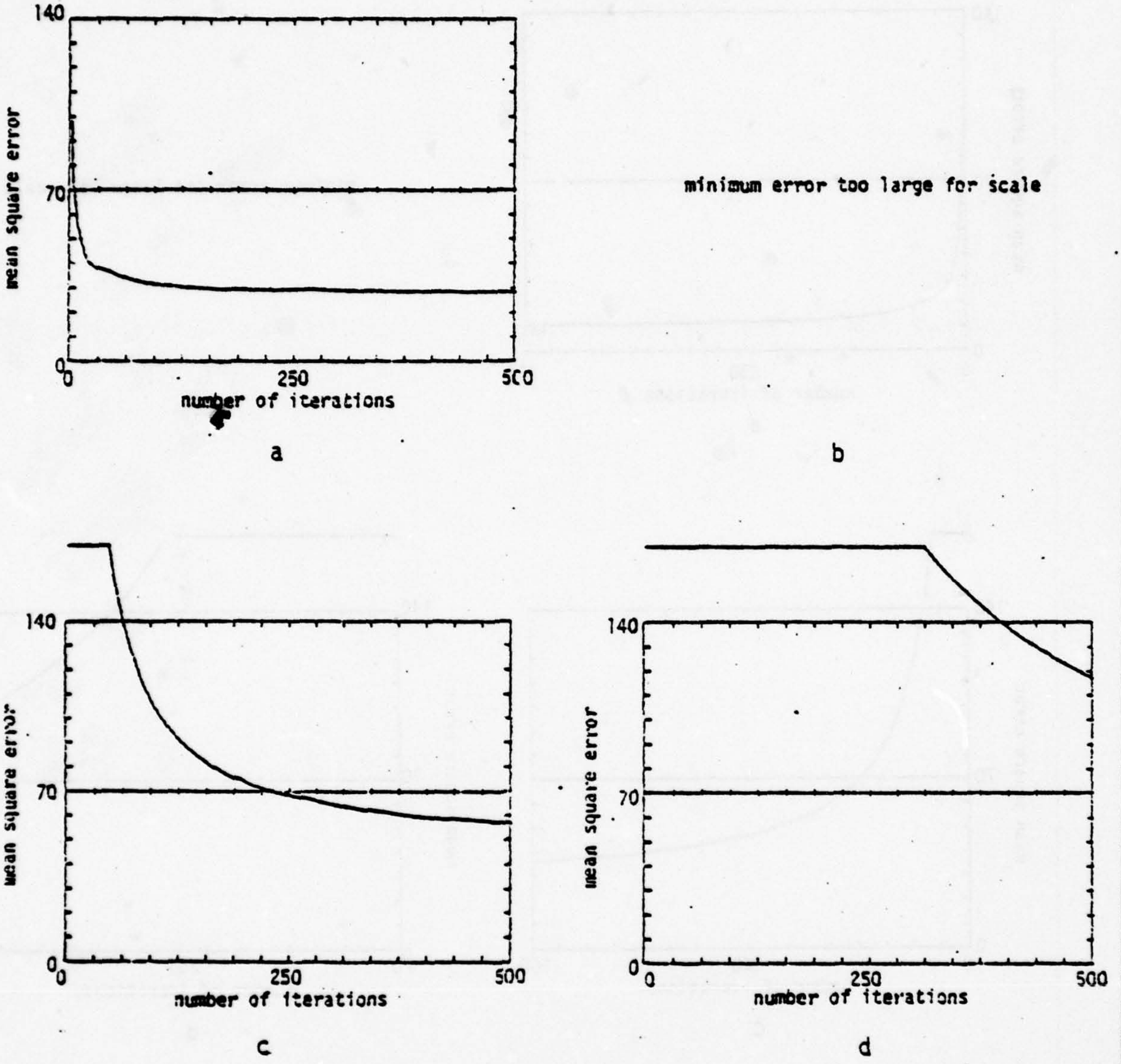


Figure 3.6. Computer Simulation of the Two Input System.
S/N = 1.0 and ρ_{12} = .85.

(a) Expected value algorithm.
(b) LMS algorithm, μ = .01.
(c) LMS algorithm, μ = .001.
(d) LMS algorithm, μ = .0001.

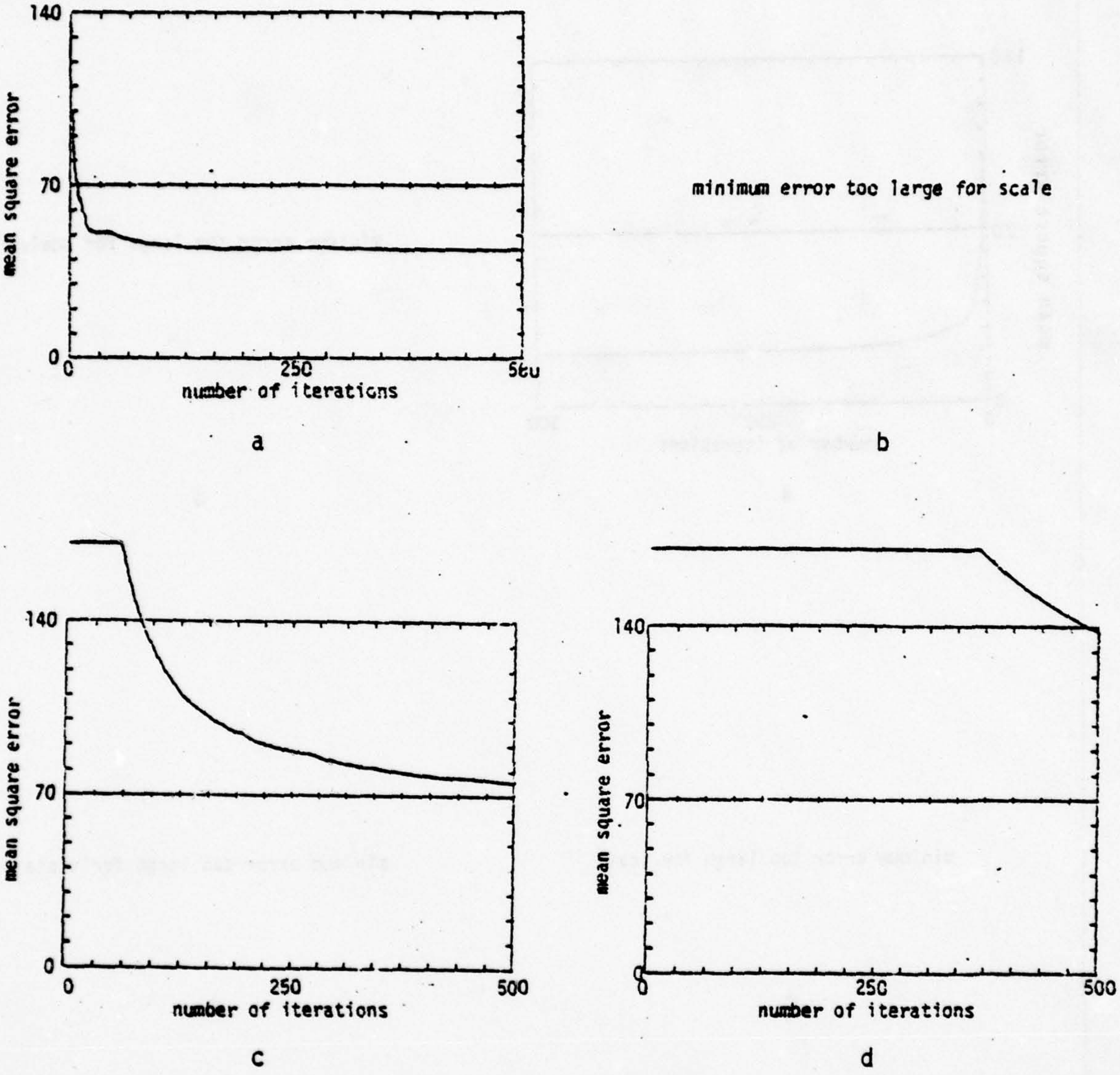
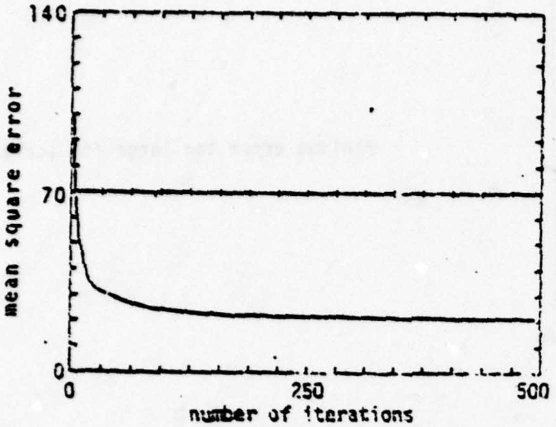


Figure 3.7. Computer Simulation of the Two Input System. S/N = 1.0 and ρ_{12} = .75.

(a) Expected value algorithm.
(b) LMS algorithm, μ = .01.
(c) LMS algorithm, μ = .001.
(d) LMS algorithm, μ = .0001.



a

minimum error too large for scale

b

minimum error too large for scale

c

minimum error too large for scale

d

Figure 3.8. Computer Simulation of the Two Input System. S/N = .10 and ρ_{12} = .99.

(a) Expected value algorithm.
(b) LMS algorithm, μ = .01.
(c) LMS algorithm, μ = .001.
(d) LMS algorithm, μ = .0001.

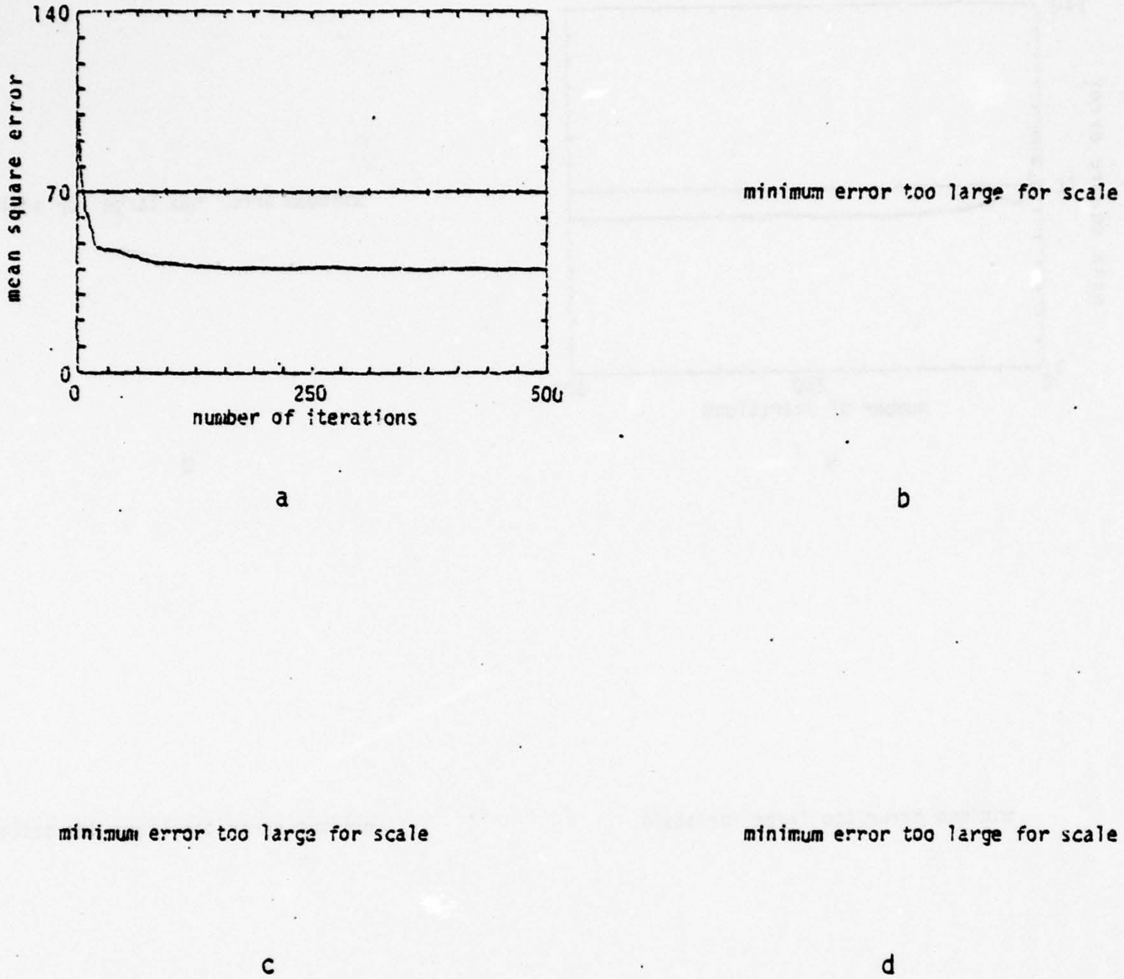
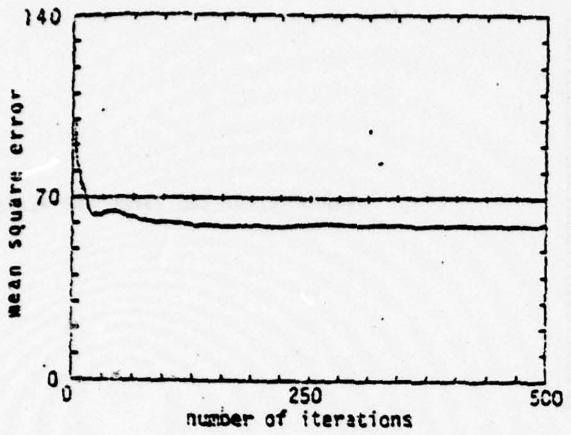


Figure 3.9. Computer Simulation of the Two Input System.
S/N = .10 and ρ = .98.

(a) Expected value algorithm.
(b) LMS algorithm, μ = .01.
(c) LMS algorithm, μ = .001.
(d) LMS algorithm, μ = .0001.



a

minimum error too large for scale

b

minimum error too large for scale

c

minimum error too large for scale

d

Figure 3.10. Computer Simulation of the Two Input System. S/N = .10 and ρ_{12} = .97.

(a) Expected value algorithm.
(b) LMS algorithm, μ = .01.
(c) LMS algorithm, μ = .001.
(d) LMS algorithm, μ = .0001.

CHAPTER IV

EXTENSION TO THREE INPUT NOISE CANCELING SYSTEM: DERIVATION OF THE ALGORITHMS

The algorithms derived and simulated in Chapters II and III are extended in this chapter to a three input noise canceling system. The goal of the algorithms is to minimize the mean square error. Again the performance of the algorithms is determined from a computer simulation.

A three input noise canceling system has one input containing a mixture of signal and noise and two inputs of noise only. The weight vector contains two weights. The three input system is shown in Figure 4.1.

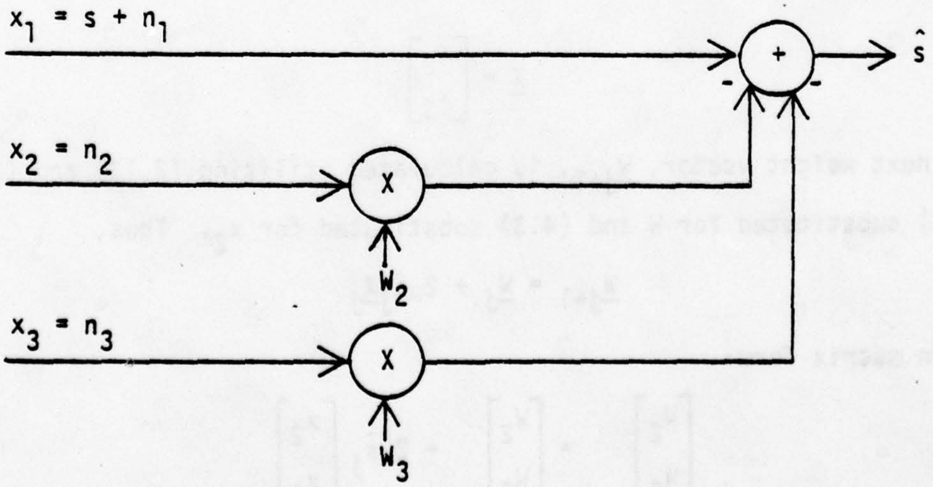


Figure 4.1. Three Input Noise Canceling System.

The signal, s, is uncorrelated with the noises, n_1, n_2, and n_3, but the noises are correlated with each other. The estimated signal, $\hat{s}$, is given by

$$\hat{s} = x_1 - W_2 x_2 - W_3 x_3 \tag{4.1}$$

and the error is defined by (2.2) as

$$\varepsilon = \hat{s} - s \tag{2.2}$$

The weights, W_2 and W_3, are determined such that the signal estimate, $\hat{s}$, is a linear least mean square estimate.

1. <u>LMS Algorithm</u>

The LMS algorithm is easily extended to the three input noise canceling system of Figure 4.1 because several scalar quantities of the two input system are simply replaced with vectors. The weight, W, of the two input system is replaced with the vector $\underline{W}$ where

$$\underline{W} = \begin{bmatrix} W_2 \\ W_3 \end{bmatrix} \tag{4.2}$$

The input x_2 is replaced with a vector of noise only inputs:

$$\underline{X} = \begin{bmatrix} x_2 \\ x_3 \end{bmatrix} \tag{4.3}$$

The next weight vector, $\underline{W}_{j+1}$, is calculated utilizing (2.13) and (2.59) with (4.2) substituted for W and (4.3) substituted for x_2. Thus,

$$\underline{W}_{j+1} = \underline{W}_j + 2\mu \hat{s}_j \underline{X}_j \tag{4.4}$$

or in matrix form:

$$\begin{bmatrix} W_2 \\ W_3 \end{bmatrix}_{j+1} = \begin{bmatrix} W_2 \\ W_3 \end{bmatrix}_j + 2\mu \hat{s}_j \begin{bmatrix} x_2 \\ x_3 \end{bmatrix}_j \tag{4.5}$$

where

$$\hat{s}_j = x_{1_j} - \underline{W}_j^T \underline{X}_j \qquad (4.6)$$

or

$$\hat{s}_j = x_{1_j} - W_{2_j} x_{2_j} - W_{3_j} x_{3_j} \qquad (4.7)$$

In terms of the inputs, the weights of the three input system are given in Figure 4.2.

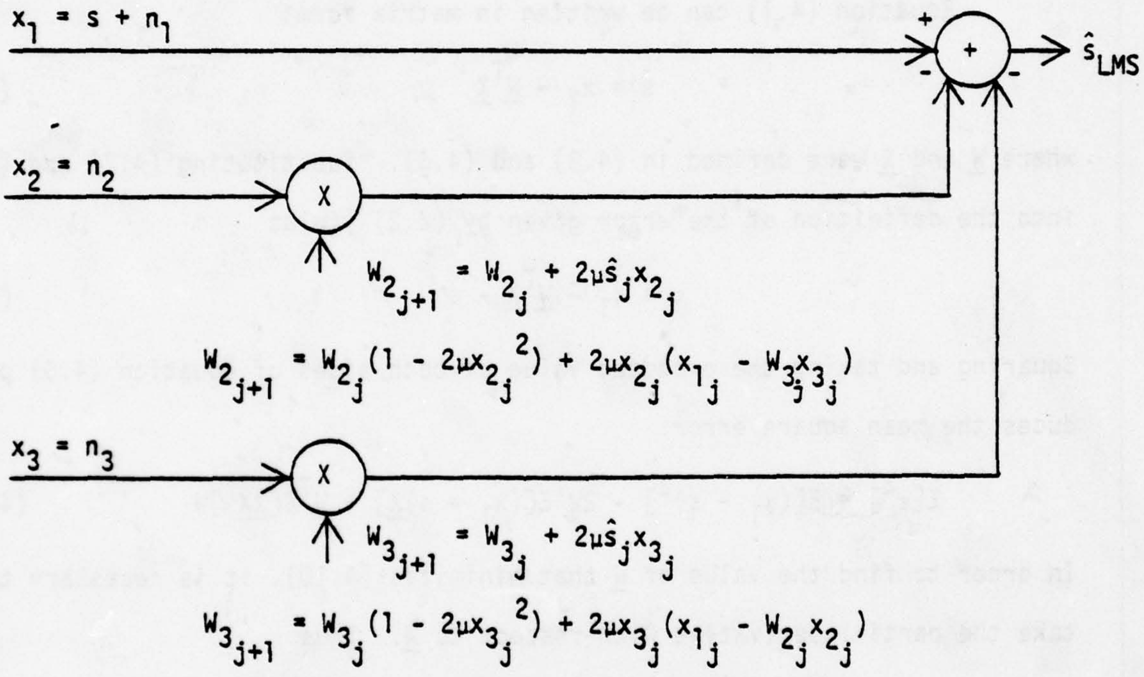

$x_1 = s + n_1$
+
+
-
-
$\hat{s}_{LMS}$
$x_2 = n_2$
X
$W_{2_{j+1}} = W_{2_j} + 2\mu\hat{s}_j x_{2_j}$
$W_{2_{j+1}} = W_{2_j}(1 - 2\mu x_{2_j}^{2}) + 2\mu x_{2_j}(x_{1_j} - W_{3_j} x_{3_j})$
$x_3 = n_3$
X
$W_{3_{j+1}} = W_{3_j} + 2\mu\hat{s}_j x_{3_j}$
$W_{3_{j+1}} = W_{3_j}(1 - 2\mu x_{3_j}^{2}) + 2\mu x_{3_j}(x_{1_j} - W_{2_j} x_{2_j})$

Figure 4.2. LMS Algorithm.

The LMS algorithm can be easily extended to any number of inputs necessary to solve a noise canceling problem. The expected value algorithm involves a matrix inversion, but is an estimation of the theoretical optimum solution. The tradeoff between the two algorithms is in terms of the complexity and performance.

2. Expected Value Algorithm

The expected value algorithm attempts to minimize the mean square error by estimating the optimum weight vector for Figure 4.1. The optimum weight vector can be determined by solving for the weights that make the partial derivative of the mean square error with respect to the weights equal to zero. This is equivalent to solving the Wiener-Hopf equation for a three input system.

Equation (4.1) can be written in matrix form:

$$\hat{s} = x_1 - \underline{W}^T\underline{X} \tag{4.8}$$

where $\underline{W}$ and $\underline{X}$ were defined in (4.2) and (4.3). Substituting (4.2) and (4.3) into the definition of the error given by (2.2) yields

$$\varepsilon = x_1 - \underline{W}^T\underline{X} - s \tag{4.9}$$

Squaring and taking the expected value of both sides of equation (4.9) produces the mean square error:

$$E[\varepsilon^2] = E[(x_1 - s)^2] - 2\underline{W}^T E[(x_1 - s)\underline{X}] + \underline{W}^T E[\underline{XX}^T]W \tag{4.10}$$

In order to find the value of $\underline{W}$ that minimizes (4.10), it is necessary to take the partial derivative with respect to $\underline{W}$. Thus

$$\frac{\partial}{\partial \underline{W}} E[\varepsilon^2] = -2E[(x_1 - s)\underline{X}] + 2E[\underline{XX}^T]W \tag{4.11}$$

Expanding (4.11) and recalling the hypotheses that $E[sx_2] = 0$ and $E[sx_3] = 0$ gives

$$\frac{\partial}{\partial \underline{W}} E[\varepsilon^2] = -2E[x_1\underline{X}] + 2E[\underline{XX}^T]\underline{W} \tag{4.12}$$

Define the two expected value terms in (4.12) as follows:

$$\underline{\phi_{XX}} = E[\underline{XX}^T]$$

or

$$\underline{\Phi}_{\underline{XX}} = \begin{bmatrix} E[x_2x_2] & E[x_2x_3] \\ E[x_2x_3] & E[x_3x_3] \end{bmatrix} \tag{4.13}$$

and

$$\underline{\Phi}_{x_1\underline{X}} = E[x_1\underline{X}]$$

or

$$\underline{\Phi}_{x_1\underline{X}} = \begin{bmatrix} E[x_1x_2] \\ E[x_1x_3] \end{bmatrix} \tag{4.14}$$

Thus, substituting (4.13) and (4.14) into (4.12) gives

$$\frac{\partial}{\partial \underline{W}} E[\varepsilon^2] = -2\underline{\Phi}_{x_1\underline{X}} + 2\underline{\Phi}_{\underline{XX}}\underline{W} \tag{4.15}$$

In order to find the minimum error, the partial derivative in (4.15) is set equal to zero. Solving for $\underline{W}$ gives

$$\underline{\Phi}_{\underline{XX}}\underline{W} = \underline{\Phi}_{x_1\underline{X}} \tag{4.16}$$

Equation (4.16) can also be written in matrix form utilizing (4.13) and (4.14):

$$\begin{bmatrix} E[x_2x_2] & E[x_2x_3] \\ E[x_3x_2] & E[x_3x_3] \end{bmatrix} \begin{bmatrix} W_2 \\ W_3 \end{bmatrix} = \begin{bmatrix} E[x_1x_2] \\ E[x_1x_3] \end{bmatrix} \tag{4.17}$$

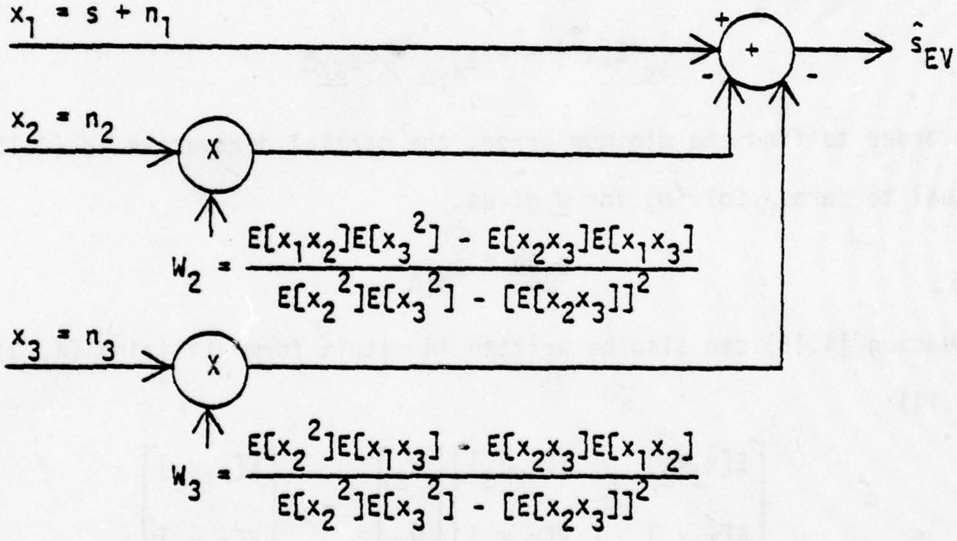
Taking the matrix inverse, (4.17) can be solved for W_2 and W_3:

$$W_2 = \frac{E[x_1x_2]E[x_3x_3] - E[x_2x_3]E[x_1x_3]}{E[x_2x_2]E[x_3x_3] - E[x_2x_3]E[x_2x_3]} \tag{4.18}$$

and

$$W_3 = \frac{E[x_2x_2]E[x_1x_3] - E[x_2x_3]E[x_1x_2]}{E[x_2x_2]E[x_3x_3] - E[x_2x_3]E[x_2x_3]} \tag{4.19}$$

Thus the values of W_2 and W_3 as determined in (4.18) and (4.19) are the optimum values necessary to minimize the mean square error. They are in fact the solutions to the Wiener-Hopf equation, (1.2), for a three input system where $\underline{R} = \underline{\phi}_{\underline{XX}}$ and $\underline{P} = \underline{\phi}_{x_1\underline{X}}$. The expected value algorithm utilizes these optimum weights in order to estimate the signal. The expected values are mathematical terms and are estimated (see (2.61)) for the computer simulation of the three input system. Figure 4.3 shows the three input noise canceling system with the weights determined from the theoretical derivation of the expected value algorithm.



$x_1 = s + n_1$
+
+
-
-
$\hat{s}_{EV}$
$x_2 = n_2$
X
$W_2 = \dfrac{E[x_1x_2]E[x_3^2] - E[x_2x_3]E[x_1x_3]}{E[x_2^2]E[x_3^2] - [E[x_2x_3]]^2}$
$x_3 = n_3$
X
$W_3 = \dfrac{E[x_2^2]E[x_1x_3] - E[x_2x_3]E[x_1x_2]}{E[x_2^2]E[x_3^2] - [E[x_2x_3]]^2}$

Figure 4.3. Expected Value Algorithm.

Since the expected value and LMS algorithms both converge to the same error (see Appendix A), it is useful to determine the theoretical minimum mean square error.

3. Theoretical Minimum Mean Square Error

In order to effectively compare the two algorithms, it is desirable to determine the theoretical minimum mean square error. The optimum weight vector for a linear least mean square estimate was determined by the expected value algorithm. Thus using (4.18) and (4.19) for the weight vector, (4.1) for the estimated signal, and (2.40) for the minimum error, the theoretical minimum error can be calculated:

$$\varepsilon_{min} = E[(\hat{s} - s)^2] \tag{2.40}$$

and

$$\varepsilon_{MMS} = E[(n_1 - W_2x_2 - W_3x_3)^2] \tag{4.20}$$

Equation (4.20) can be simplified using Figure 4.1 and replacing x_1, x_2, and x_3 with their equivalent values in terms of s, n_1, n_2, and n_3. The definition of correlation given by (2.34) is also used:

$$\varepsilon_{MMS} = E[n_1{}^2] - 2W_2E[n_1n_2] - 2W_3E[n_1n_3] + 2W_2W_3E[n_2n_3] + W_2{}^2E[n_2{}^2] + W_3{}^2E[n_3{}^2]$$

and thus

$$\varepsilon_{MMS} = E[n_1{}^2] - 2W_2\rho_{12}\sqrt{E[n_1{}^2]E[n_2{}^2]} - 2W_3\rho_{13}\sqrt{E[n_1{}^2]E[n_3{}^2]} + 2W_2W_3\rho_{23}\sqrt{E[n_2{}^2]E[n_3{}^2]} + W_2{}^2E[n_2{}^2] + W_3{}^2E[n_3{}^2] \tag{4.21}$$

The optimum weight vector given by (4.18) and (4.19) can be rewritten in terms of s, n_1, n_2, and n_3. The definition of correlation in (2.34) also helps to simplify the vector. Thus,

$$W_2 = c_1\sqrt{\frac{E[n_1{}^2]}{E[n_2{}^2]}} \tag{4.22}$$

where

$$c_1 = \frac{\rho_{12} - \rho_{23}\rho_{13}}{1 - \rho_{23}{}^2} \tag{4.23}$$

and

$$W_3 = c_2\sqrt{\frac{E[n_1^2]}{E[n_3^2]}} \tag{4.24}$$

where

$$c_2 = \frac{\rho_{13} - \rho_{12}\rho_{23}}{1 - \rho_{23}^2} \tag{4.25}$$

Substituting (4.22) and (4.24) into (4.21), the theoretical minimum error can be simplified further:

$$\epsilon_{MMS} = E[n_1^2][1 - 2\rho_{12}c_1 - 2\rho_{13}c_2 + 2c_1c_2\rho_{23} + c_1^2 + c_2^2] \tag{4.26}$$

where c_1 and c_2 are defined by (4.23) and (4.25).

Since the weight vectors of both algorithms converge to the optimum weight vector (see Appendix A), the theoretical minimum mean square error of both algorithms is given by (4.26).

4. <u>Summary of the Algorithms</u>

The algorithms are summarized here in the forms that are used for the computer simulation. The inputs, weights, and estimates are presented in terms of the jth iteration.

A. <u>LMS Algorithm</u>

The LMS algorithm utilized the computed value of $\hat{s}_{LMS}$ for the computer simulation. The values of the weights at each iteration are stored so that they can be used to compute the weights for the next iteration. Therefore the LMS algorithm is simulated for the jth iteration as shown in Figure 4.4.

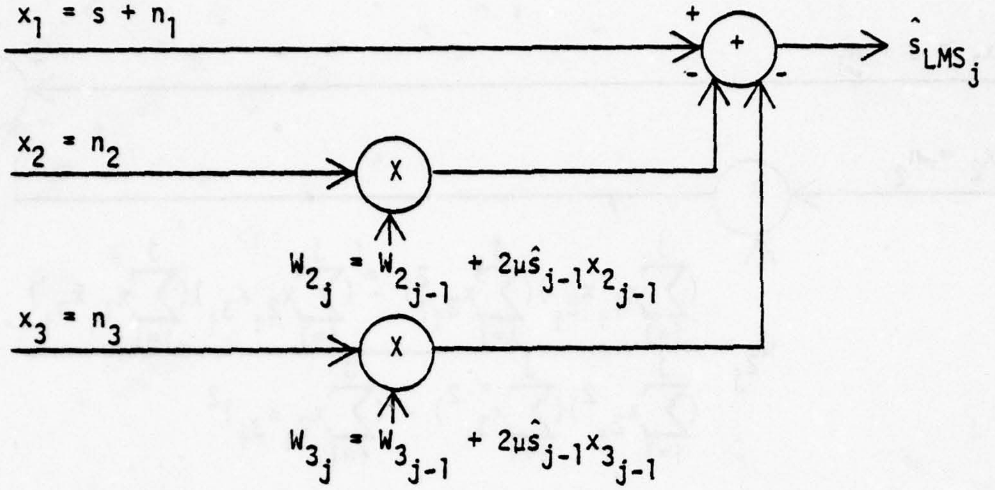

$x_1 = s + n_1$
+
+
-
-
$\hat{s}_{LMS_j}$
$x_2 = n_2$
X
$W_{2_j} = W_{2_{j-1}} + 2\mu\hat{s}_{j-1}x_{2_{j-1}}$
$x_3 = n_3$
X
$W_{3_j} = W_{3_{j-1}} + 2\mu\hat{s}_{j-1}x_{3_{j-1}}$

Figure 4.4. LMS Algorithm as Computer Simulated.

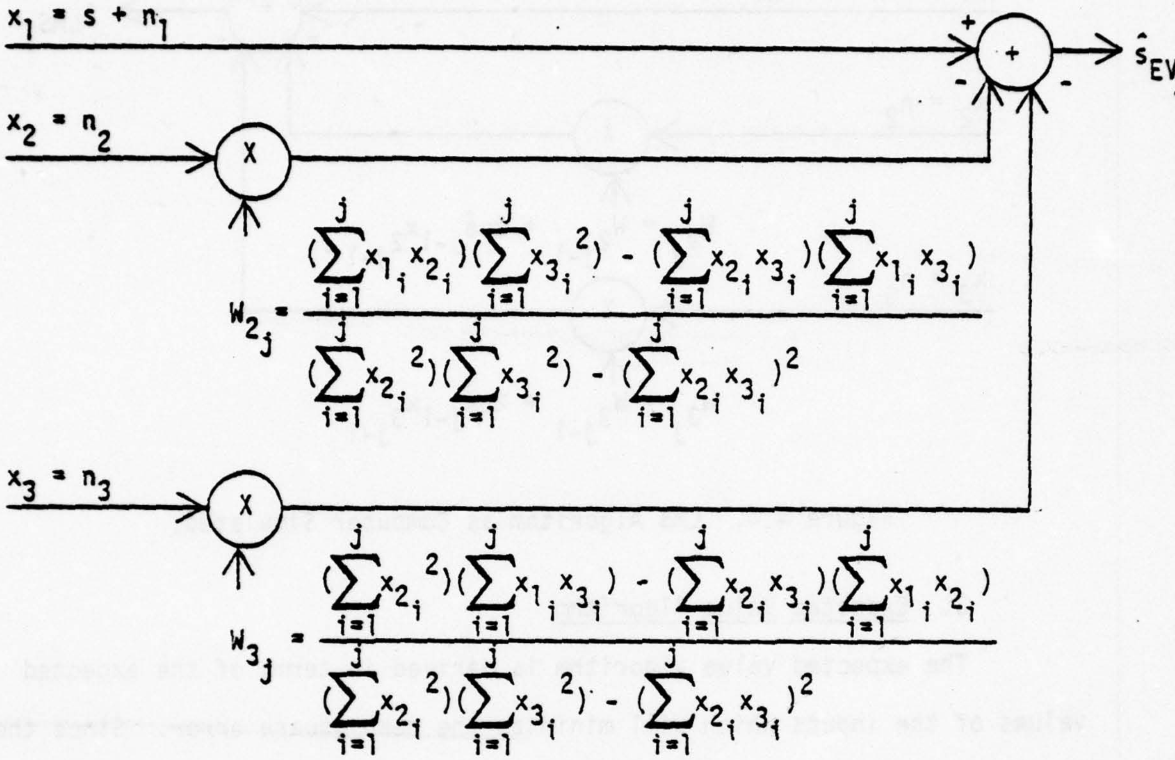
B. Expected Value Algorithm

The expected value algorithm is derived in terms of the expected values of the inputs which will minimize the mean square error. Since the expected values are mathematical quantities which are not known exactly, the expected value algorithms must estimate them and is therefore only an approximation to the optimum solution.

The expected values were estimated using the iterative procedure described in Chapter II. The expected value algorithm for the jth iteration of the computer simulation is implemented as seen in Figure 4.5.

C. Comparison of the Algorithms

The optimum weight vector will produce the best results for the three input system. However, since it can't be calculated directly the expected value algorithm approximates it by estimating the expected values. As in the two input case, the LMS algorithm attempts to estimate the optimum weight vector as one term while the expected value algorithm estimates each expected value before combining them to form the weight vector. Although this method



$$W_{2_j} = \frac{\left(\sum_{i=1}^{j} x_{1_i} x_{2_i}\right)\left(\sum_{i=1}^{j} x_{3_i}{}^2\right) - \left(\sum_{i=1}^{j} x_{2_i} x_{3_i}\right)\left(\sum_{i=1}^{j} x_{1_i} x_{3_i}\right)}{\left(\sum_{i=1}^{j} x_{2_i}{}^2\right)\left(\sum_{i=1}^{j} x_{3_i}{}^2\right) - \left(\sum_{i=1}^{j} x_{2_i} x_{3_i}\right)^2}$$

$$W_{3_j} = \frac{\left(\sum_{i=1}^{j} x_{2_i}{}^2\right)\left(\sum_{i=1}^{j} x_{1_i} x_{3_i}\right) - \left(\sum_{i=1}^{j} x_{2_i} x_{3_i}\right)\left(\sum_{i=1}^{j} x_{1_i} x_{2_i}\right)}{\left(\sum_{i=1}^{j} x_{2_i}{}^2\right)\left(\sum_{i=1}^{j} x_{3_i}{}^2\right) - \left(\sum_{i=1}^{j} x_{2_i} x_{3_i}\right)^2}$$

Figure 4.5. Expected Value Algorithm as Computer Simulated.

is more complex and more time consuming due to the increased number of multiplications and divisions, the improved performance justifies it. The LMS algorithm could also be written in terms of a feedback system as in the two input case, but this is not necessary. The expected value algorithm can't be a feedback system because it needs the jth inputs to compute the jth estimate. The LMS algorithm instead computes the jth weights with the data through the (j-1)st inputs. This method is not as effective as that of the expected value algorithm.

CHAPTER V

COMPUTER SIMULATION OF THE THREE INPUT NOISE CANCELING SYSTEM

The three input algorithms developed in Chapter IV were simulated on a PDP-11/45 computer at Duke University. Figures 4.4 and 4.5 give the algorithms in terms of their computer implementation.

1. Problem Description

Parameter values had to be chosen before starting the simulation. The signal was a dc signal with value 10.0 as in the two input case. The values of the initial weight vector for the LMS algorithm were again constant. The first weight, W_2, was set equal to 5.0 as in the two input case, and the second, W_3, was arbitrarily set to 3.0. These values were not important due to the large number of iterations.

The choice of the parameter μ for the LMS algorithm was again a problem as it was for the two input system. In order for the algorithm to converge, the value of μ must be in the range (see Appendix A) defined as:

$$|1 - 2\mu E[\underline{X}^T\underline{X}]| < 1 \qquad (5.1)$$

where $\underline{X}$ is defined by (4.3). The three μ values of .01, .001, and .0001, corresponding to graphs b, c, and d, respectively on each page of figures, are the same as those used in the two input system simulation. Also for convenience in the simulation,

$$E[n_1^2] = E[n_2^2] = E[n_3^2] \qquad (5.2)$$

When the signal-to-noise ratio is 10.0, all three values of μ are within the range for convergence of the algorithm. When the signal-to-noise ratio is lowered to 1.0, the μ value of .01 is outside the range of convergence. Both μ values of .01 and .001 are outside the range of convergence when the signal-to-noise ratio is .1. For a given signal-to-noise ratio, a decrease in the parameter μ results in a decrease in the rate of convergence since the quantity $|1-2\mu E[\underline{X}^T\underline{X}]|$ approaches one.

The noise was generated using the method described in Appendix B. Independent zero mean Gaussian random numbers were generated and then transformed using this method to correlated Gaussian random numbers. The correlations, ρ_{12}, ρ_{13}, and ρ_{23}, were parameter values of the program. The input matrix $E[\underline{X}_j\underline{X}_j^T]$ has to be a positive semi-definite matrix. Thus the transformation matrix, made up of correlation values and given by (5.3)

$$\begin{bmatrix} 1 & \rho_{12} & \rho_{13} \\ \rho_{12} & 1 & \rho_{23} \\ \rho_{13} & \rho_{23} & 1 \end{bmatrix} \qquad (5.3)$$

has to positive semi-definite. This restricts the choices of the correlation values. Once two correlation values have been chosen, the third must be selected in such a manner that the matrix is positive semi-definite.

The mean square error was approximated using (3.2) as in the two input case. Iterative estimates were used for the expected values as shown in Figure 4.5.

The theoretical minimum mean square error was defined in (4.26). Its value depends only upon the values of $E[n_1^2]$, ρ_{12}, ρ_{13}, and ρ_{23}. When the three correlations all have values of zero, the worst case is realized and no noise can be effectively canceled. As the correlations increase in value, noise canceling occurs.

The method of comparison of the two algorithms was the same as for the two input case. Graphs of the mean square error as a function of the number of iterations were used. Twenty-five, five hundred iteration ensembles were averaged to obtain the curve that was plotted. As in the two input system, the three input system simulation curves are compared for differences in rates of convergence and minimum error after 500 iterations.

2. Performance Comparison

Following are the graphs of the mean square error as a function of the number of iterations. Each page contains graphs with the same signal-to-noise ratio and correlation values. One graph using the expected value algorithm and graphs of three values of μ for the LMS algorithm are on each page. Again all of the graphs are plotted on the same scale. The straight line effect (see Appendix C) is still observed when the value of the mean square error is too large for the boundaries of the scale. Table 5.1 gives representative values for the curves.

Figures 5.1 through 5.3 set the signal-to-noise ratio to 10.0. Each figure has a different set of correlation values. The expected value algorithm exhibits the fastest rate of convergence and the smallest error after five hundred iterations. However, graphs 5.1b and 5.2b, which are the μ values of .01 for the LMS algorithm, are very similar to those of the expected value algorithm. Thus if a "correct" μ value is chosen, the results can be close to the optimum results as estimated by the expected value algorithm. As the μ value decreases, a slower rate of convergence is evidenced and the minimum error after five hundred iterations is further from the theoretical minimum. If two of the correlation values are small (Figure 5.3), the difference between the expected value and LMS algorithm becomes much greater.

Figures 5.4 through 5.6 lower the signal-to-noise ratio to 1.0. The same correlation values are used again. The smaller signal-to-noise ratio produces dramatic results. The expected value algorithm generates the best curves in terms of the rate of convergence and the minimum error. The μ value of .001 for the LMS algorithm produces the only curve that can truly be compared with the expected value algorithm. Nevertheless, its rate of convergence is much slower and its minimum error is much greater. The μ value of .01 generally produces values that are too large for the desired scale or curves that appear to converge to a large error. For this signal-to-noise ratio, the μ value of .01 is outside the range necessary for convergence. The μ value of .0001 has such a slow rate of convergence that the straight line effect is again observed. As the correlation decreases, this case of the LMS algorithm has an even slower rate of convergence and eventually the value of the mean square error for five hundred iterations is too large for the desired scale.

Figures 5.7 through 5.9 lower the signal-to-noise ratio to .10. Only the expected value algorithm generates curves with values small enough for the desired scale. The μ values of .01 and .001 are outside the range for convergence of the algorithm. When μ equals .0001, the rate of convergence is too slow for convergence within five hundred iterations and the minimum values of the curves are still too large for the scale. Although the μ value of .00001 is not shown, the minimum value was found to be larger (for only five hundred iterations) than the case where μ equaled .0001. Again there is a storage and graphing problem created due to the large numbers generated by the LMS algorithm that is not present with the expected value algorithm.

3. Summary of the Three Input System Results

The expected value algorithm proved to be the superior algorithm for the three input case as it was for the two input system. The rate of convergence is consistently faster and the minimum error is smaller than for any of the μ values of the LMS algorithm that were tested. For a signal-to-noise ratio of 10.0, the LMS algorithm produces curves similar to those of the expected value algorithm. However, this occurs for only one particular μ value since other μ values do not produce results that are as similar. As the signal-to-noise ratio decreases, the expected value algorithm clearly produces the best results. Several μ values also generate curves which have values too large to be plotted.

In several cases, the expected value algorithm generated curves whose minimum values were less than the theoretical minimum values. This is probably a peculiarity of the computer simulation and the computer used. Since all of the noise was generated with the same variance, this probably contributed to the small minimum errors. This would be especially noticeable with small signal-to-noise ratios because of a higher possibility of arithmetic roundoff errors.

The expected value algorithm is more complex in terms of the number of multiplications and divisions required to implement it. Again, the several estimate approach of the weight vector used by the expected value algorithm enables a better estimate in fewer iterations. For a three input noise canceling system, a two dimensional matrix inverse is required. Although an inverse of this size matrix is more complex than the LMS algorithm, the tradeoff in terms of performance appears to justify this added complexity.

The effect of changes in correlation is hard to assertain because of the large number of values per curve. Generally as the correlations were lowered, the difference between performance of the two algorithms became greater. With three inputs, the signal-to-noise ratio appeared to have more of an effect on the results.

The problem with the LMS algorithm is the same as in the two input system. Values for the initial weight vectors and for μ must be chosen. It is almost impossible to determine beforehand the best values for these parameters (in terms of minimum mean square error and convergence rate). Unfortunately, unless the μ value is chosen correctly, the LMS algorithm may not converge or may have a slow rate of convergence and large errors even after five hundred iterations.

Table 5.1. Representative Values of the Curves Obtained: Three Input Noise Canceling System

Each page contains four graphs labeled a, b, c, and d. These have characteristic values given by several parameters:

$\frac{S}{N}$ - the signal-to-noise ratio

ρ_{12} - the correlation between the noises n_1 and n_2

ρ_{13} - the correlation between the noises n_1 and n_3

ρ_{23} - the correlation between the noises n_2 and n_3

TMIN - the theoretical minimum mean square error

Algorithm - the algorithm employed-either expected value or LMS

μ - the value of the parameter μ for the LMS algorithm

MAX - the maximum value of the curve

MIN - the minimum value of the curve

Figure	$\frac{S}{N}$	ρ_{12}	ρ_{13}	ρ_{23}	TMIN	Algorithm	μ	MAX	MIN
5.1a	10	.90	.90	.80	1.0	Expected value		100.0	2.390
b						LMS	.01	72.29	6.361
c						LMS	.001	72.29	11.12
d						LMS	.0001	72.29	46.04
5.2a	10	.75	.20	.75	1.371	Expected value		100.0	2.933
b						LMS	.01	66.57	4.557
c						LMS	.001	67.04	13.83
d						LMS	.0001	68.50	50.13
5.3a	10	.25	.99	.25	.1989	Expected value		99.99	2.893
b						LMS	.01	207.1	48.55
c						LMS	.001	207.1	16.02
d						LMS	.0001	207.1	65.10
5.4a	1.0	.90	.90	.80	10.0	Expected value		99.95	2.882
b						LMS	.01	911.1	538.9
c						LMS	.001	722.9	20.35
d						LMS	.0001	722.9	104.9
5.5a	1.0	.75	.20	.75	13.71	Expected value		99.99	5.148
b						LMS	.01	665.7	108.4
c						LMS	.001	665.7	21.56
d						LMS	.0001	670.3	132.0

Table 5.1 cont'd

Figure	$\frac{S}{N}$	ρ_{12}	ρ_{13}	ρ_{23}	TMIN	Algorithm	μ	MAX	MIN
5.6a	1.0	.25	.99	.25	1.989	Expected value		100.4	7.386
b						LMS	.01	Overflow	Overflow
c						LMS	.001	2071	77.33
d						LMS	.0001	2071	139.3
5.7a	.10	.90	.90	.80	100.0	Expected value		99.32	7.724
b						LMS	.01	Overflow	Overflow
c						LMS	.001	9189	799.9
d						LMS	.0001	7229	157.6
5.8a	.10	.75	.20	.75	137.1	Expected value		100.1	27.06
b						LMS	.01	Overflow	Overflow
c						LMS	.001	6657	299.9
d						LMS	.0001	6657	191.5
5.9a	.10	.25	.99	.25	19.89	Expected value		100.9	52.08
b						LMS	.01	Overflow	Overflow
c						LMS	.001	Overflow	Overflow
d						LMS	.0001	2.071E04	359.3

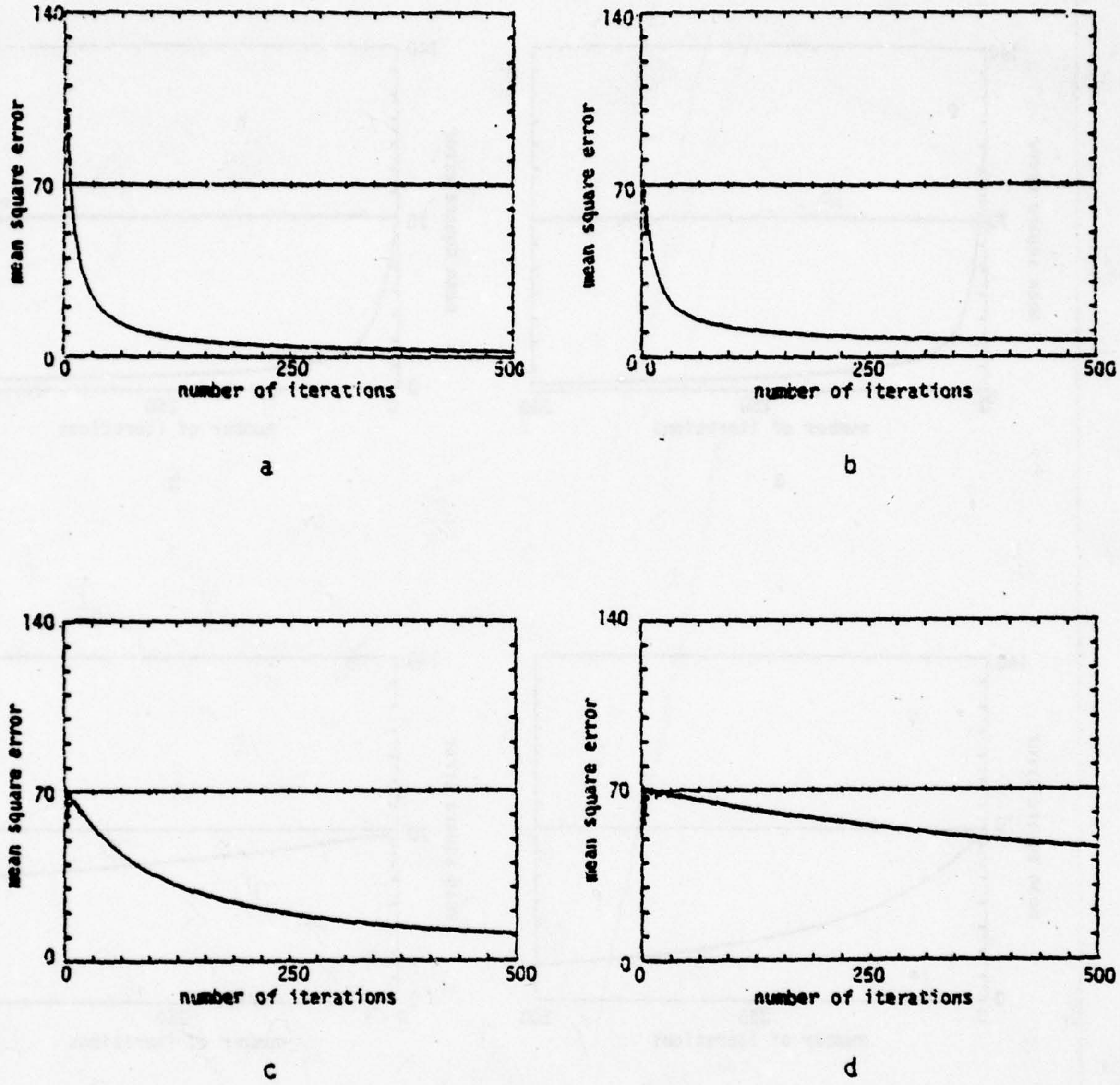


Figure 5.1. Computer Simulation of the Three Input System. S/N = 10.0, ρ_{12} = .90, ρ_{13} = .90, and ρ_{23} = .80.

(a) Expected value algorithm.
(b) LMS algorithm, μ = .01.
(c) LMS algorithm, μ = .001.
(d) LMS algorithm, μ = .0001.

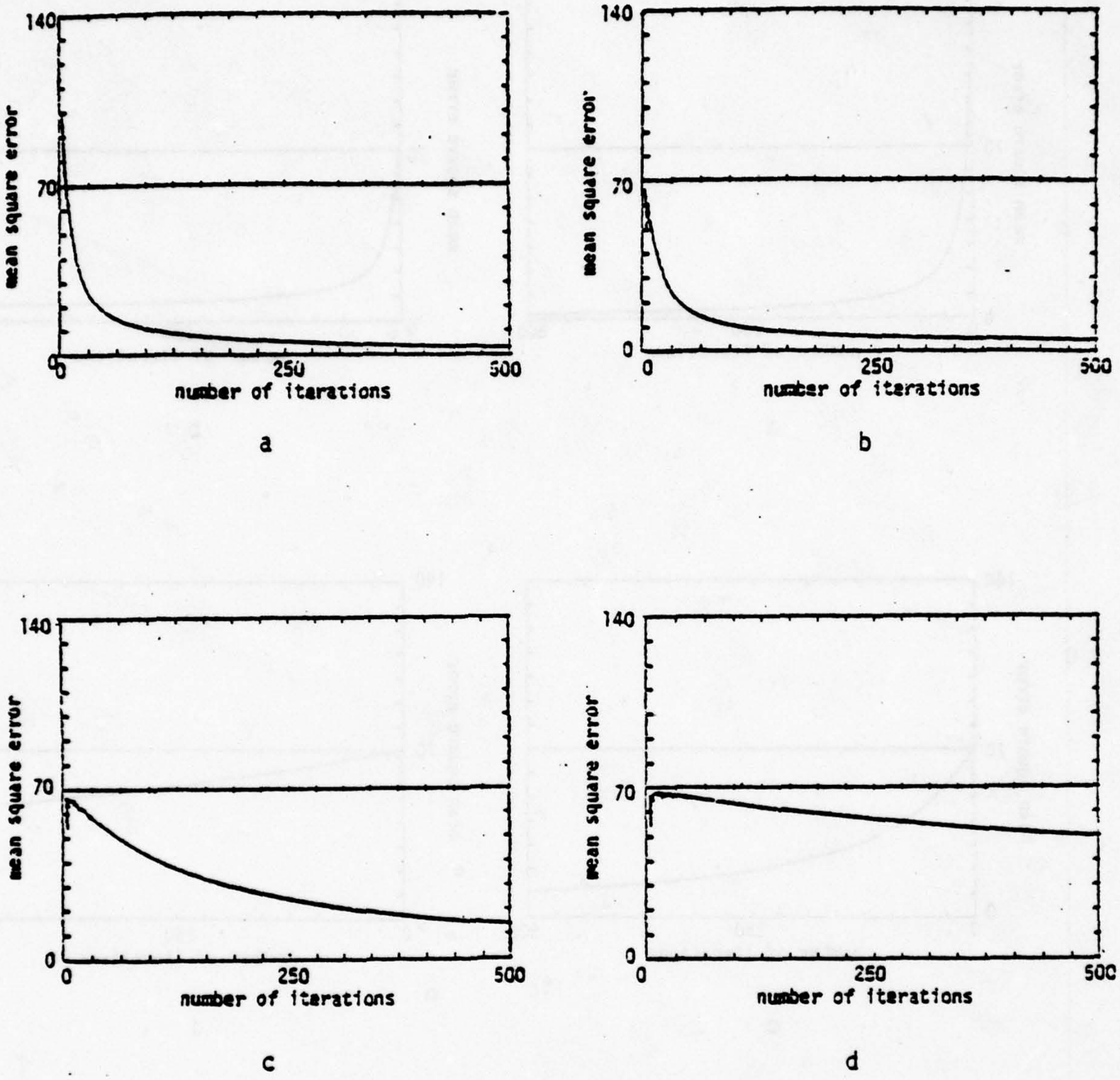


Figure 5.2. Computer Simulation of the Three Input System. S/N = 10.0, ρ_{12} = .75, ρ_{13} = .20, and ρ_{23} = .75.

(a) Expected value algorithm.
(b) LMS algorithm, μ = .01.
(c) LMS algorithm, μ = .001.
(d) LMS algorithm, μ = .0001.

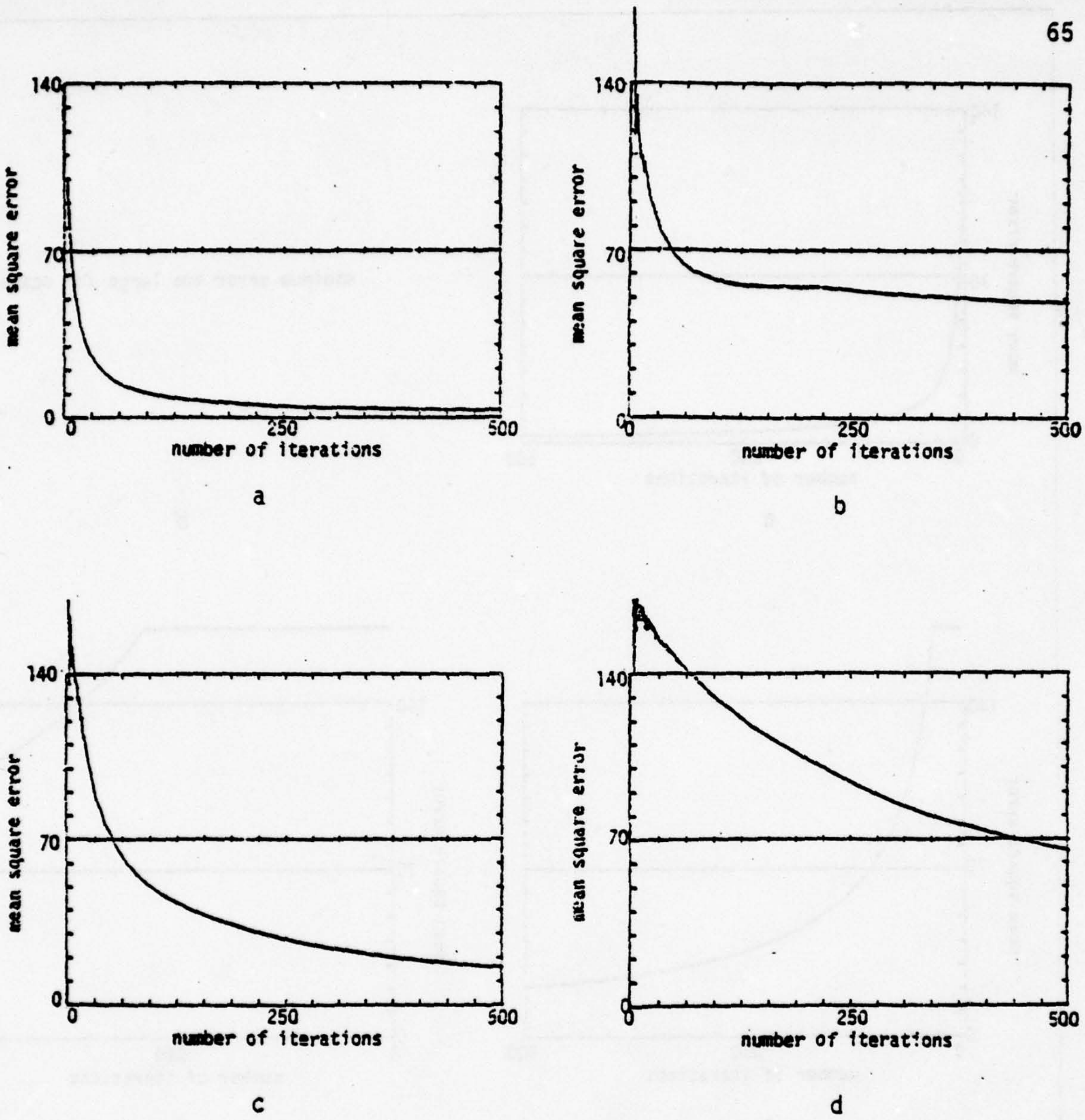


Figure 5.3. Computer Simulation of the Three Input System. S/N = 10.0, ρ_{12} = .25, ρ_{13} = .99, and ρ_{23} = .25.

(a) Expected value algorithm.
(b) LMS algorithm, μ = .01.
(c) LMS algorithm, μ = .001.
(d) LMS algorithm, μ = .0001.

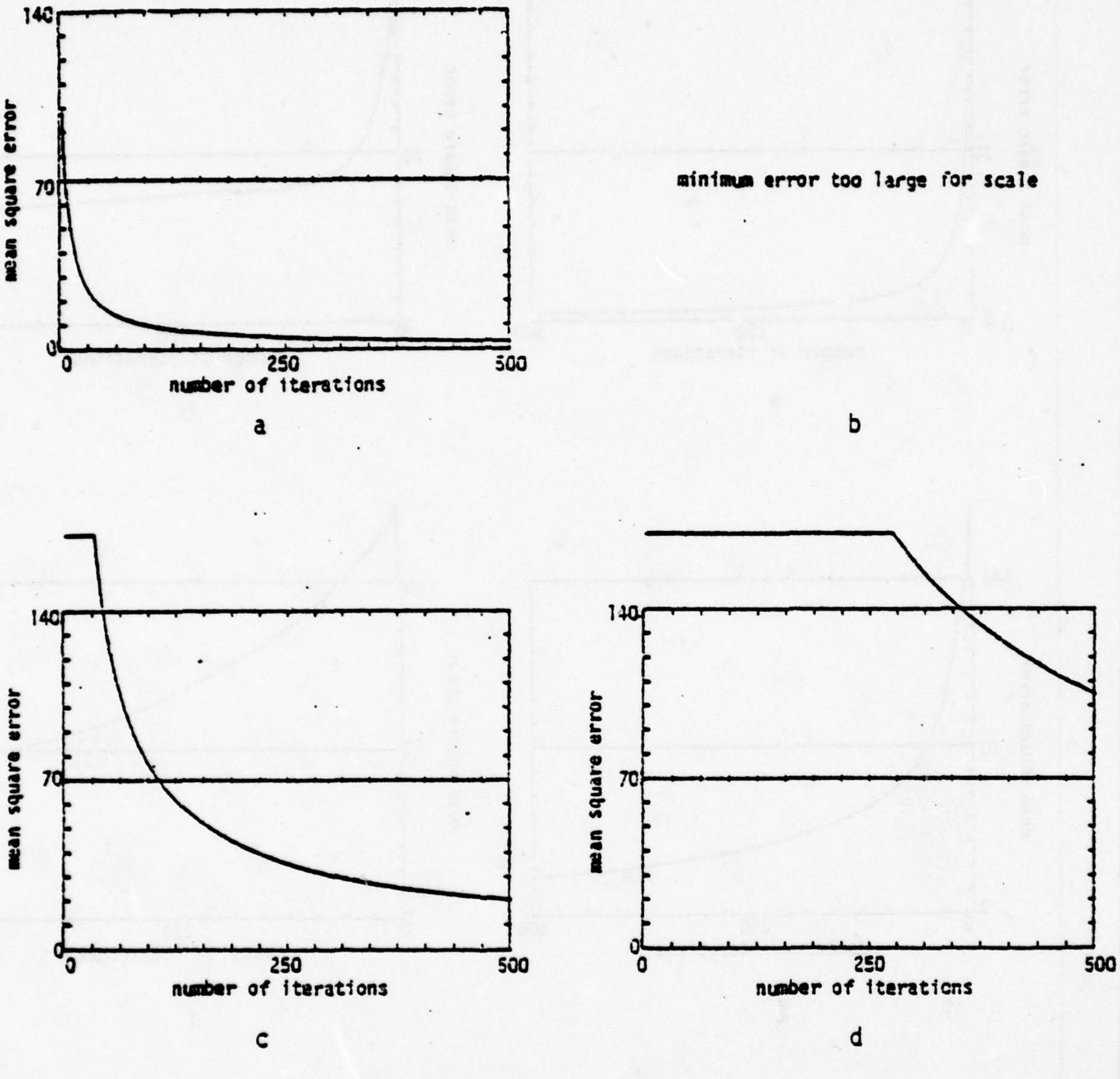


Figure 5.4. Computer Simulation of the Three Input System. S/N = 1.0, ρ_{12} = .90, ρ_{13} = .90, and ρ_{23} = .80.

(a) Expected value algorithm.
(b) LMS algorithm, μ = .01.
(c) LMS algorithm, μ = .001.
(d) LMS algorithm, μ = .0001.

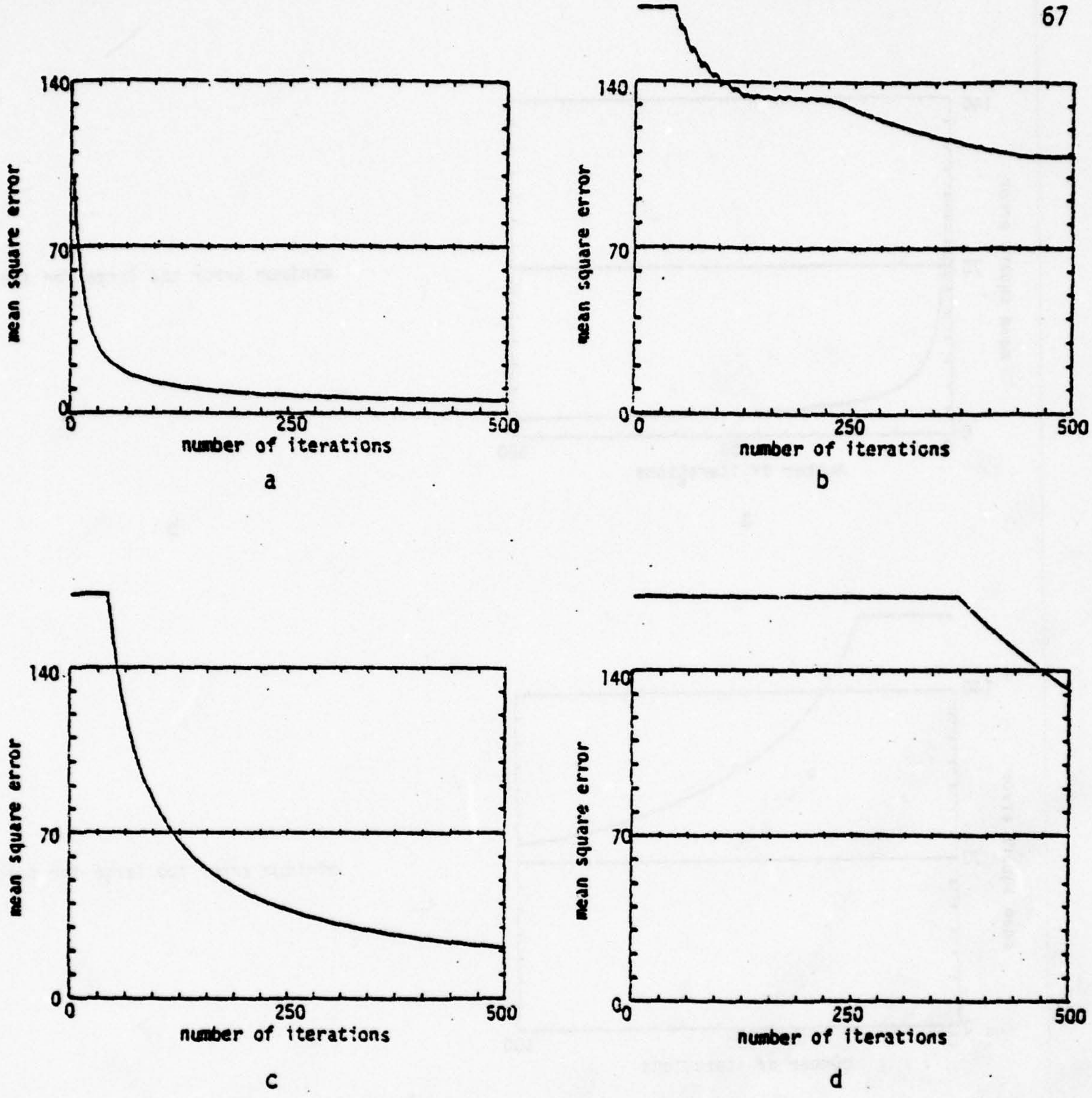


Figure 5.5. Computer Simulation of the Three Input System. S/N = 1.0, ρ_{12} = .75, ρ_{13} = .20, and ρ_{23} = .75.

(a) Expected value algorithm.
(b) LMS algorithm, μ = .01.
(c) LMS algorithm, μ = .001.
(d) LMS algorithm, μ = .0001.

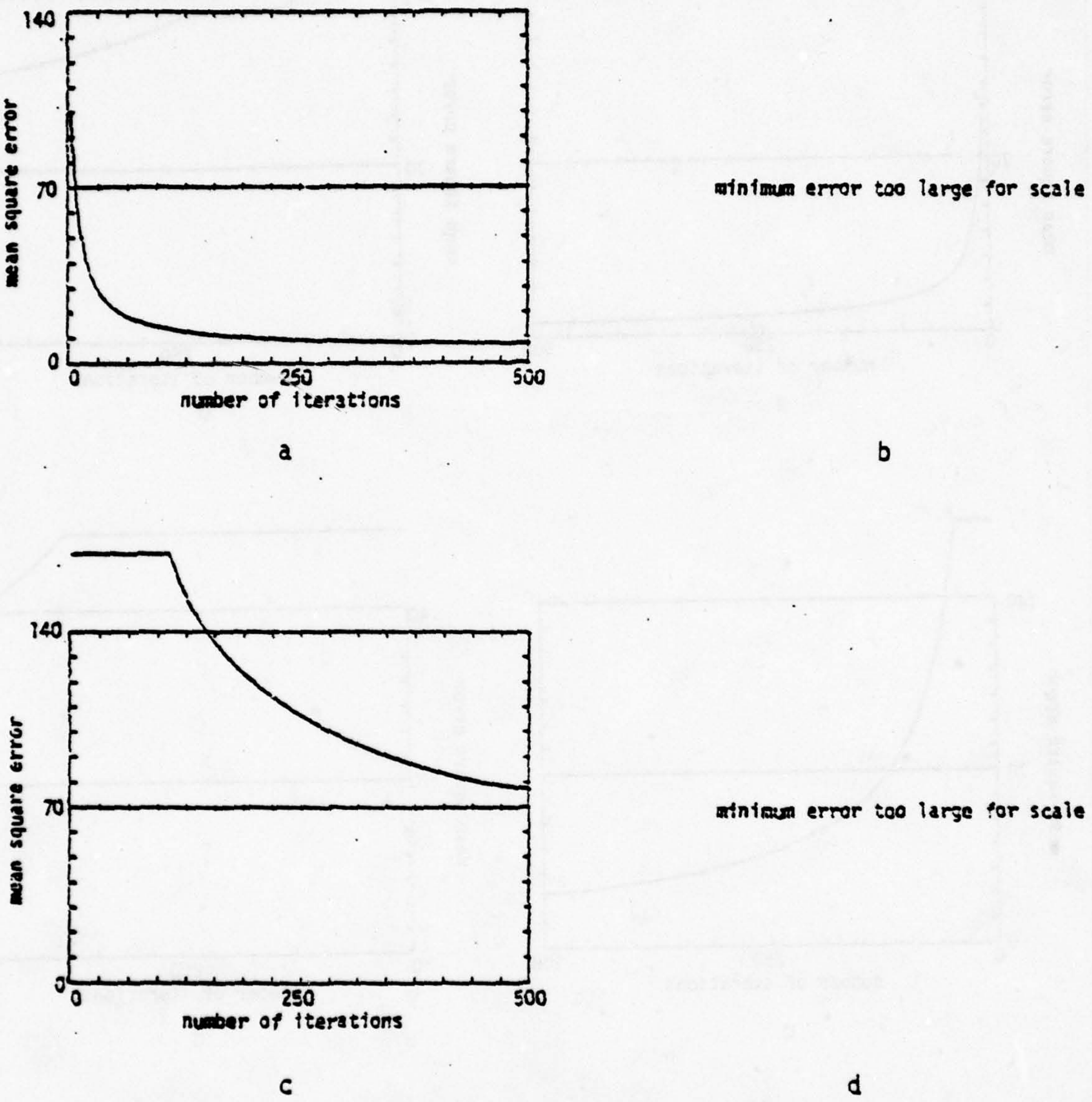
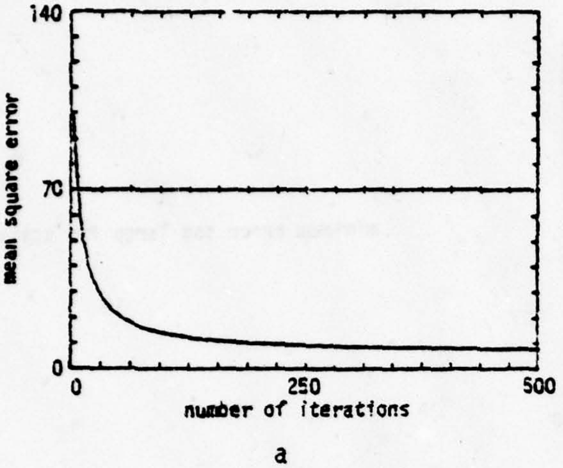


Figure 5.6. Computer Simulation of the Three Input System. S/N = 1.0, ρ_{12} = .25, ρ_{13} = .99, and ρ_{23} = .25.

(a) Expected value algorithm.
(b) LMS algorithm, μ = .01.
(c) LMS algorithm, μ = .001.
(d) LMS algorithm, μ = .0001.



minimum error too large for scale

b

minimum error too large for scale

c

minimum error too large for scale

d

Figure 5.7. Computer Simulation of the Three Input System. S/N = .10, ρ_{12} = .90, ρ_{13} = .90, and ρ_{23} = .80.

(a) Expected value algorithm.
(b) LMS algorithm, μ = .01.
(c) LMS algorithm, μ = .001.
(d) LMS algorithm, μ = .0001.

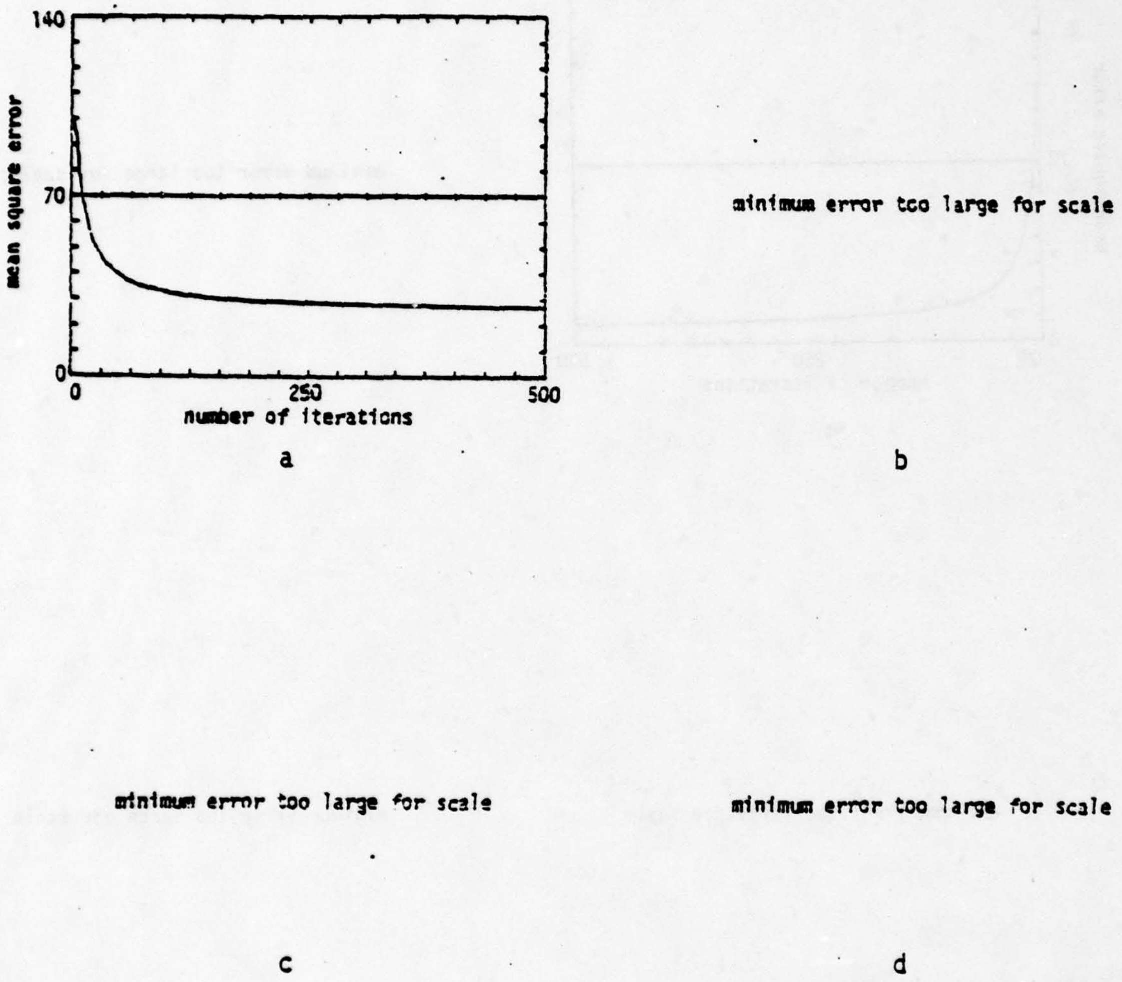
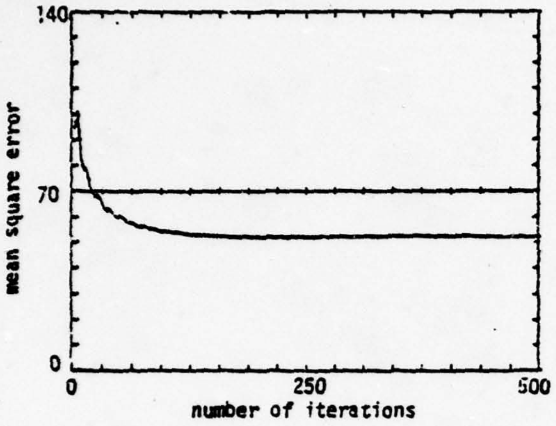


Figure 5.8. Computer Simulation of the Three Input System.
$S/N = .10$, $\rho_{12} = .75$, $\rho_{13} = .20$, and $\rho_{23} = .75$.

(a) Expected value algorithm.
(b) LMS algorithm, $\mu = .01$.
(c) LMS algorithm, $\mu = .001$.
(d) LMS algorithm, $\mu = .0001$.



a

minimum error too large for scale

b

minimum error too large for scale

c

minimum error too large for scale

d

Figure 5.9. Computer Simulation of the Three Input System. S/N = .10, ρ_{12} = .25, ρ_{13} = .99, and ρ_{23} = .25.

(a) Expected value algorithm.
(b) LMS algorithm, μ = .01.
(c) LMS algorithm, μ = .001.
(d) LMS algorithm, μ = .0001.

CHAPTER VI

SUMMARY AND FUTURE WORK

1. Summary

This research compares two algorithms used in noise canceling systems. The first part studies a two input noise canceling system. The expected value and LMS (least mean square) algorithms are compared in terms of their mean square error performance. The second section extends both algorithms to a three input noise canceling system. These results are then compared.

Chapter II derives the algorithms for the two input noise canceling system. The optimum weight vector is given by the Wiener-Hopf equation as explained in Chapter I. The expected value algorithm as implemented for the computer simulation is an estimate to the optimum algorithm in that the mathematical expected values are replaced with estimates. The LMS algorithm attempts to lessen the complexity involved in the matrix inversion by using gradient techniques and the method of steepest descent.

Chapter III reveals the results of the two input system computer simulation. The expected value algorithm proved superior in that it had a faster rate of convergence and lower minimum errors after five hundred iterations. The difference was especially apparent at lower signal-to-noise ratios and lower values of noise correlation. The LMS algorithm also presented problems in that values for the initial weight vector and for μ had to be chosen. Although the choice of the initial weight vector was not as important when considering five hundred iterations, the value of μ was extremely important. In order for the algorithm to converge, the value of the term had to be within a specified range which proved small when only five hundred iterations were considered.

The algorithms are extended to a three input noise canceling system in Chapter IV. Again the expected value algorithm is an estimate to the optimum for a minimum linear least mean square estimate. The LMS algorithm is much simpler because it does not involve a matrix inversion.

Chapter V reveals the results of the computer simulation of the three input system. The results are similar to those for the two input system. The expected value algorithm has a faster rate of convergence and smaller minimum errors. However, for several cases of correlation values, the LMS algorithm produced results that were extremely similar to those produced by the expected value algorithm. Nevertheless, this only happened for one value of μ while the other two μ values tested produced results that were much worse. The additional complexity of the expected value algorithm due to the matrix inverse is still justified. Unless the "correct" μ value is chosen, the results will be much worse than the optimum.

In general, the expected value algorithm performs better than the LMS algorithm. For a noise canceling system with only two or three inputs, an estimate to the optimum method is better. With a small number of inputs, the added complexity involved in inverting the input matrix is justified in terms of performance. This added complexity in forming the weight vector is probably the reason for the better performance. Although both algorithms compute the same terms (in terms of the data), the expected value algorithm updates several estimates before combining these to form one estimate for the weight vector. The LMS algorithm, however, only forms one estimate. The several estimate approach of the expected value algorithm produces a better estimate of the weight vector in a fewer number of iterations.

The additional problems encountered when implementing the LMS algorithm (that is, choosing values for μ and the initial weight vector) also discourage its use. The expected value algorithm does not have these problems and its performance is better in terms of rate of convergence and minimum errors.

2. Future Research

This research should encourage efforts to study the tradeoffs between complexity and performance that exist in many algorithms. Often algorithms are developed for the more intricate cases of a problem. Thus while the algorithm may be the only practical way to solve the more complex problem, this is not always true when the problem is in a simpler form.

The two input noise canceling system was studied thoroughly and the expected value algorithm produced better results. The three input system needs to be studied more carefully. The relationship between changes in correlation values should be established. With variations for three correlation values, only a few were selected to study here. The importance of one correlation value with respect to the others should be ascertained. Changes in the variance between the noise inputs should also be studied.

In general, the importance of the initial weight vector of the LMS algorithm should be studied. For this research, the value was arbitrarily chosen. If less than five hundred iterations are essential, the initial weight value will be more important.

This study should also be extended to systems with more inputs to determine how large a system can be handled practically with the expected value algorithm. Many programs today are available to invert a matrix quickly. This would enable larger systems to employ this estimate to the optimum weight vector.

This research is only a beginning, but it should encourage further research in this area. The major contribution should be the questioning of algorithms to solve simpler problems when the optimum method would produce much better results.

APPENDICES

APPENDIX A

CONVERGENCE OF THE WEIGHT VECTOR FOR THE LMS ALGORITHM

In order to make the comparison between the expected value and LMS algorithms valid, it is necessary that their weight vectors converge to the same value. The weight vector for the expected value algorithm is shown to be an estimate to the optimum weight vector in Section 3 of Chapter II. Therefore, it is only necessary to prove that the weight vector for the LMS algorithm converges to the optimum weight vector.

Looking first at the two input noise canceling system, the weight vector for the LMS algorithm is defined to be

$$W_{LMS_{j+1}} = W_{LMS_j} + 2\mu\hat{s}_j x_{2_j} \qquad \text{(A.1)}$$

The optimum and theoretical expected value algorithms have a weight vector given by

$$W_{opt_j} = W_{EV_j} = \frac{E[x_{1_j} x_{2_j}]}{E[x_{2_j}^2]} \qquad \text{(A.2)}$$

To prove that the LMS weight vector converges to the optimum weight vector, it is essential to show that

$$\lim_{j \to \infty} E[W_{LMS_{j+1}}] = W_{opt} \qquad \text{(A.3)}$$

For the two input noise canceling system,

$$\hat{s}_j = x_{1_j} - W_j x_{2_j} \tag{A.4}$$

Substituting (A.1) and (A.4) into (A.3) produces

$$E[W_{LMS_{j+1}}] = E[(1 - 2\mu x_{2_j}^{\;2})W_{LMS_j} + 2\mu x_{1_j} x_{2_j}] \tag{A.5}$$

Reducing the index by one, W_{LMS_j} can be written in terms of $W_{LMS_{j-1}}$. Thus (A.5) becomes

$$E[W_{LMS_{j+1}}] = E[(1-2\mu x_{2_j}^{\;2})(1-2\mu x_{2_{j-1}}^{\;2})W_{LMS_{j-1}} + (1-2\mu x_{2_j}^{\;2})(2\mu x_{1_{j-1}} x_{2_{j-1}}) + 2\mu x_{1_j} x_{2_j}] \tag{A.6}$$

Equation (A.6) can be rewritten in terms of W_{LMS_0}. Both inputs x_1 and x_2 are uncorrelated over time. Therefore, the expected value can be taken of each product in (A.6):

$$E[W_{LMS_{j+1}}] = (1-2\mu E[x_2^{\;2}])^{j+1} W_{LMS_0} + 2\mu E[x_1 x_2][1 + (1-2\mu E[x_2^{\;2}]) + (1-2\mu E[x_2^{\;2}])^2 + \ldots + (1-2\mu E[x_2^{\;2}])^j] \tag{A.7}$$

Recognizing that the second term in (A.7) is a geometric series, (A.7) can be simplified:

$$E[W_{LMS_{j+1}}] = (1-2\mu E[x_2^{\;2}])^{j+1} W_{LMS_0} + \left[\frac{1-[1-2\mu E[x_2^{\;2}]]^{j+1}}{1-[1-2\mu E[x_2^{\;2}]]}\right] 2\mu E[x_1 x_2] \tag{A.8}$$

In order for (A.8) to converge, it is required that

$$|1 - 2\mu E[x_2^{\;2}]| < 1 \tag{A.9}$$

Assuming that (A.9) is true, taking the limit as j approaches infinity of both sides of (A.8) gives

$$\lim_{j \to \infty} E[W_{LMS_{j+1}}] = \frac{E[x_1 x_2]}{E[x_2^2]} \qquad (A.10)$$

This is the desired result. The weight vector for the LMS algorithm converges to the optimum weight vector when (A.9) is true.

A study of convergence was made in more detail with an arbitrary set of parameters for the two input noise canceling system. A signal-to-noise ratio of 10.0 and a correlation of .95 were used for the study. The values of μ tested were .01, .001, .0001, and .00001. Figure A.1 shows the graphs of the mean square error versus the number of iterations for one ensemble and the above parameter values. One ensemble was graphed instead of the usual average of twenty-five ensembles to emphasize the fluctuations in the error. Looking at Figure A.1, the largest μ value coincides with the fastest rate of convergence, but the mean square error does not steadily or smoothly decrease to a minimum value. Instead the value fluctuates considerably. The second graph with μ = .001 exhibits the smoothest decline towards a minimum value, but the rate of convergence is slower than the curve in (a) and the mean square error still varies with a small number of iterations. Nevertheless, the mean square error in (b) has reached a smaller value after five hundred iterations than in (a). Graph (c) displays an even slower rate of convergence than (b) and graph (d) does not appear to begin to converge after five hundred iterations.

For the above parameters, the value of the optimum weight is approximately .95. Examining the value of the weight vector at each iteration reveals more about the rate of convergence towards the optimum weight vector. Although with μ = .01, the rate of convergence is the fastest, there are large fluctuations in the value of the weight vector. Even after two thousand iterations, the weight vector does not appear to converge, but instead fluctuates in value between -1 and +3. With $\mu = .001$, the rate of convergence is slower, but the weight vector exhibits fewer variations in value. The fluctuations are mainly between .7 and 1.7. The case with $\mu = .0001$ shows an even slower rate of convergence than the previous case, but the fluctuations in the values of the weight vector have almost disappeared. Inspecting the rate of convergence for two thousand iterations indicates that the weight should converge after approximately twenty-five hundred iterations. The last case studied was with $\mu = .00001$. The weight vector again showed almost no fluctuations in value, but instead, a steady decline towards the optimum value. Convergence based on this rate will not occur until approximately six thousand iterations.

The results of the convergence test can be extended to the three input case. Although the study is not as extensive as for the two input system, it is shown here that the weight vector for the LMS algorithm converges to the optimum weight vector. It was shown in Section 2 of Chapter IV that the weight vector for the expected value algorithm is an estimate to the optimum weight vector.

The weight vector for the LMS algorithm can be written in vector form:

$$\underline{W}_{LMS_{j+1}} = \underline{W}_{LMS_j} + 2\mu\hat{s}_j\underline{X}_j \qquad (A.11)$$

or

$$\begin{bmatrix} W_2 \\ W_3 \end{bmatrix}_{LMS_{j+1}} = \begin{bmatrix} W_2 \\ W_3 \end{bmatrix}_{LMS_j} + 2\mu\hat{s}_j \begin{bmatrix} x_2 \\ x_3 \end{bmatrix}_j \qquad (A.12)$$

It is necessary that

$$\lim_{j \to \infty} E[\underline{W}_{LMS_{j+1}}] = \underline{W}_{opt} \qquad (A.13)$$

The optimum weight vector is given by (4.17) and is

$$\underline{W}_{opt} = \underline{\underline{\Phi_{XX}}}^{-1} \Phi_{x_1 \underline{X}} \qquad (A.14)$$

or

$$\underline{W}_{opt} = \begin{bmatrix} E[x_2x_2] & E[x_2x_3] \\ E[x_2x_3] & E[x_3x_3] \end{bmatrix}^{-1} \begin{bmatrix} E[x_1x_2] \\ E[x_1x_3] \end{bmatrix} \qquad (A.15)$$

Taking the expected value of both sides of (A.11) gives

$$E[\underline{W}_{LMS_{j+1}}] = E[\underline{W}_{LMS_j} + 2\mu \hat{s}_j \underline{X}_j] \qquad (A.16)$$

Substituting for $\hat{s}_j$ in (A.16) yields

$$E[\underline{W}_{LMS_{j+1}}] = E[2\mu x_{1_j} \underline{X}_j + \underline{W}_{LMS_j}[1-2\mu \underline{X}_j{}^T \underline{X}_j]] \qquad (A.17)$$

Equation (A.17) can be rewritten in terms of the (j-1)st iteration:

$$E[\underline{W}_{LMS_{j+1}}] = E[2\mu x_{1_j} \underline{X}_j + 2\mu x_{1_{j-1}} \underline{X}_{j-1}(1-2\mu \underline{X}_j{}^T \underline{X}_j)$$

$$+ \underline{W}_{LMS_{j-1}}(1-2\mu \underline{X}_{j-1}{}^T \underline{X}_{j-1})(1-2\mu \underline{X}_j{}^T \underline{X}_j)] \qquad (A.18)$$

Equation (A.18) can also be written in terms of $\underline{W}_{LMS_0}$ and the expected value can be taken of each term. Thus

$$E[\underline{W}_{LMS_{j+1}}] = W_0(1-2\mu E[\underline{X}^T\underline{X}])^{j+1} + 2\mu E[x_1\underline{X}][1 + (1-2\mu E[\underline{X}^T\underline{X})] + \ldots$$

$$+ (1-2\mu E[\underline{X}^T\underline{X}])^j] \qquad (A.19)$$

Recognizing that (A.19) contains a geometric series which will converge when

$$|1 - 2\mu E[\underline{X}^T\underline{X}]| < 1 \qquad (A.20)$$

(A.19) can be written in a simpler form:

$$E[W_{LMS_{j+1}}] = W_0(1-2\mu E[\underline{X}^T\underline{X}])^{j+1} + 2\mu E[x_1\underline{X}]\,\frac{1-[1-2\mu E[\underline{X}^T\underline{X}]]^{j+1}}{1-[1-2\mu E[\underline{X}^T\underline{X}]]} \qquad (A.21)$$

Thus when (A.20) holds, the series will converge. Taking the limit of both sides of (A.21) produces

$$\lim_{j\to\infty} E[\underline{W}_{LMS_{j+1}}] = (E[\underline{X}^T\underline{X}])^{-1}E[x_1\underline{X}] \qquad (A.22)$$

This is the desired result. Thus the weight vector of the LMS algorithm does in fact converge to the optimum weight vector for both the two and three input noise canceling systems.

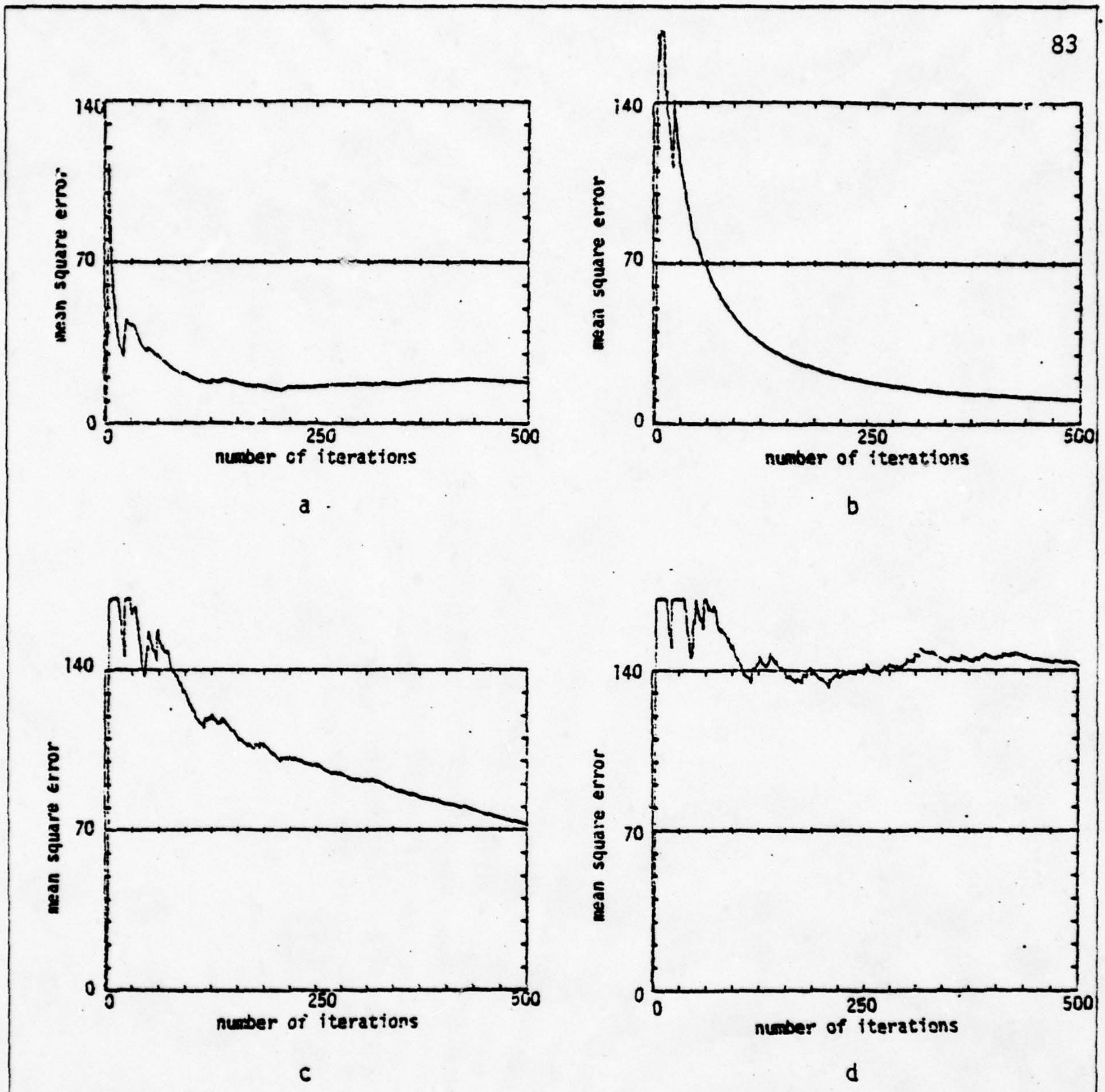


Figure A.1. Convergence Test of the LMS Algorithm for the Two Input System. S/N = 10.0 and ρ_{12} = .95.

(a) μ = .01.
(b) μ = .001.
(c) μ = .0001.
(d) μ = .00001.

APPENDIX B

NOISE GENERATION FOR THE THREE INPUT NOISE CANCELING SYSTEM

The computer simulation requires three correlated Gaussian random numbers for the noise components to the three input noise canceling system. An unpublished paper by Charles S. Liu, University of Michigan-Dearborn, develops a technique to transform a n-dimensional vector of independent Gaussian random numbers to correlated Gaussian random numbers. The method will be summarized here.

Let $\underline{Z}^T = [Z_1, Z_2, \ldots, Z_n]$ be an n-dimensional random vector whose elements are independent Gaussian random numbers with zero mean and unit variance. The vector $\underline{Z}$ may be transformed such that it contains new random variables:

$$\underline{Z}' = \underline{PZ} \tag{B.1}$$

where $\underline{P}$ is an n x n transformation matrix specified by the covariance matrix of $\underline{Z}'$:

$$\underline{Q} = E[\underline{Z}'\underline{Z}'^T]$$

$$\underline{Q} = \underline{PP}^T \tag{B.2}$$

The transformation $\underline{Z}' = \underline{PZ}$ is not unique since there are an infinite number of matrices which satisfy (B.2). Since the covariance matrix is real

AD-A055 822 DUKE UNIV DURHAM N C ADAPTIVE SIGNAL DETECTION LAB F/G 9/3
A COMPARISON OF TWO NOISE CANCELING SYSTEM ALGORITHMS. (U)
DEC 77 R P MESSINA N00014-75-C-0191
UNCLASSIFIED TR-14 NL

2 OF 2
ADA
055822

END
DATE
FILMED
8-78
DDC

symmetric, the eigenvectors, $\{\underline{Y}_i\}$ of $\underline{Q}$ are orthogonal. Thus $\underline{Q}$ can be diagonalized as shown:

$$[\underline{Y}_1,\underline{Y}_2,\ldots,\underline{Y}_n]^T\underline{Q}[\underline{Y}_1,\underline{Y}_2,\ldots,\underline{Y}_n] = \begin{bmatrix} \lambda_1 & & & 0 \\ & \lambda_2 & & \\ & & \ddots & \\ 0 & & & \lambda_n \end{bmatrix} \tag{B.3}$$

IF (B.3) is premultiplied by $[\underline{Y}_1,\underline{Y}_2,\ldots,\underline{Y}_n]$ and postmultiplied by $[\underline{Y}_1,\underline{Y}_2,\ldots,\underline{Y}_n]^T$, then

$$[\underline{Y}_1,\underline{Y}_2,\ldots,\underline{Y}_n][\underline{Y}_1,\underline{Y}_2,\ldots,\underline{Y}_n]^T\underline{Q}[\underline{Y}_1,\underline{Y}_2,\ldots\underline{Y}_n][\underline{Y}_1,\underline{Y}_2,\ldots,\underline{Y}_n]^T = \underline{Q}$$

$$= [\underline{Y}_1,\underline{Y}_2,\ldots,\underline{Y}_n]\begin{bmatrix} \lambda_1 & & & 0 \\ & \lambda_2 & & \\ & & \ddots & \\ 0 & & & \lambda_n \end{bmatrix}[\underline{Y}_1,\underline{Y}_2,\ldots,\underline{Y}_n]^T \tag{B.4}$$

Thus $\underline{Q}$ can be rewritten using (B.4) so that

$$\underline{Q} = [\underline{Y}_1,\underline{Y}_2,\ldots,\underline{Y}_n]\begin{bmatrix} \sqrt{\lambda_1} & & & 0 \\ & \sqrt{\lambda_2} & & \\ & & \ddots & \\ 0 & & & \sqrt{\lambda_n} \end{bmatrix}\begin{bmatrix} \sqrt{\lambda_1} & & & 0 \\ & \sqrt{\lambda_2} & & \\ & & \ddots & \\ 0 & & & \sqrt{\lambda_n} \end{bmatrix}[\underline{Y}_1,\underline{Y}_2,\ldots,\underline{Y}_n]^T \tag{B.5}$$

Recognizing that (B.5) is of the same form as (B.2), the transformation matrix $\underline{P}$ can be defined as

$$P = [\underline{Y}_1,\underline{Y}_2,\ldots,\underline{Y}_n]\begin{bmatrix} \sqrt{\lambda_1} & & & 0 \\ & \sqrt{\lambda_2} & & \\ & & \ddots & \\ 0 & & & \sqrt{\lambda_n} \end{bmatrix} \tag{B.6}$$

Therefore the eigenvalues and eigenvectors of the covariance matrix determine the transformation matrix necessary to obtain correlated Gaussian random numbers from independent ones. That transformation is defined by (B.1) where P, the transformation matrix, is given by (B.6).

APPENDIX C

STRAIGHT LINE EFFECT OF GRAPHS

In order to make the comparison between the algorithms clearer and more apparent, it was desirable to plot all of the mean square error curves using the same graph scale. The graphs were produced on a computer screen of limited size. When the values were too large to be within the range of the scale, the curve was plotted as a straight line at the top of the screen. Although the straight line section of the curve tends to imply that the curve has a constant value over this range, this is not the case. Figure C.1 shows this effect by rescaling one graph so that the entire curve lies within the boundaries of the scale. This is done using the maximum and minimum values of the curve as the maximum and minimum points of the scale. If a graph is replaced by the statement that the minimum error is too large for the scale, this means that for five hundred iterations the error is always greater than 140.

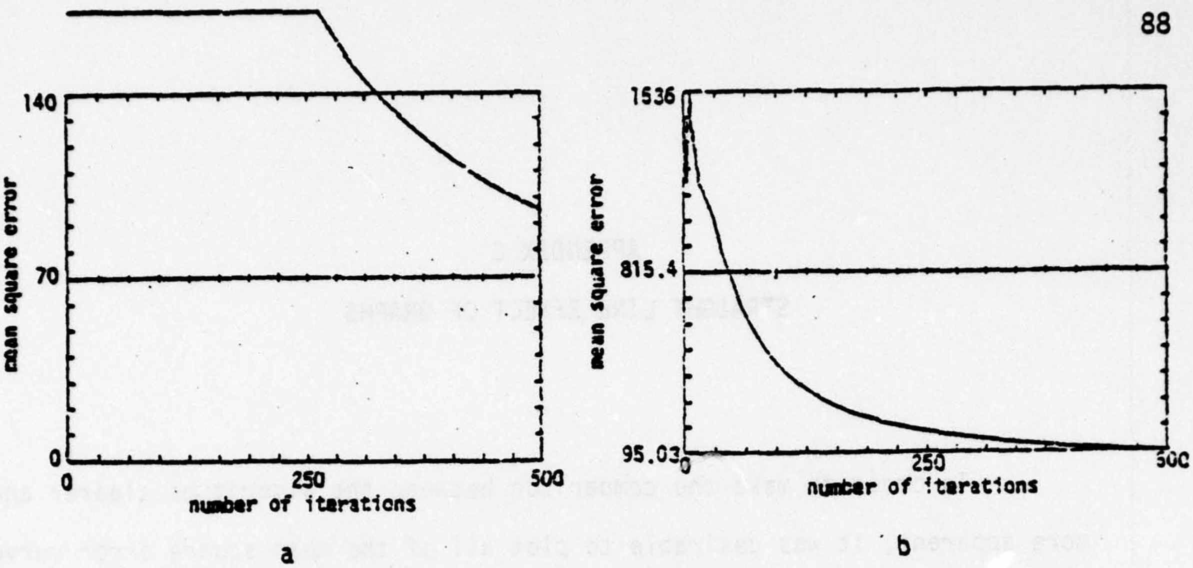


Figure C.1. Straight Line Effect (Two Input System).
S/N = 1.0 and ρ_{12} = .95.

(a) LMS algorithm, μ = .0001, normal scale
(b) LMS algorithm, μ = .0001, rescaled.

APPENDIX D
COMPUTER SIMULATION PROGRAMS

A. Two Input Noise Canceling System

Purpose:

The computer simulation was the basis of comparison between the expected value and LMS algorithms. The equations given for the algorithms in Figures 2.12 and 2.13 were implemented on a PDP-11/45 at Duke University. The mean square error was computed and plotted as a function of the number of iterations.

Specifications:

The specifications of the problem were described in Section 1 of Chapter III and they are summarized here.

1. The signal is a dc signal with value 10.0.
2. The noise was generated using the computer's uniformly distributed random number generator and transforming to zero mean Gaussian random numbers with the Box-Muller equations.
3. The initial weight vector, W(1), for the LMS algorithm was set equal to 5.0.
4. The mean square error was computed using an iterative estimate:

$$E[(\hat{s}_j - s_j)^2] \doteq \frac{\sum_{i=1}^{j}(\hat{s}_i - s_i)^2}{j} \tag{D.1}$$

5. The mean square error is computed for 500 iterations to produce one ensemble curve. Twenty-five ensembles are averaged point by point and this curve is plotted as a function of the number of iterations.

Program Structure:

A separate program is employed for each algorithm. The program ESTSIG implements the expected value algorithm and the program ANEST implements the LMS algorithm. Each program is divided into two main parts. One part generates the noise, computes the weight vector, and determines the mean square error. The other part averages the ensembles and plots the result. Figures D.1 and D.2 show the structure of the programs.

Program Segment Descriptions

1. ESTSIG -- Main program to implement the expected value algorithm and compute the mean square error; calls subroutines CGAUSS and PLOTIT.

   INPUTS: RO -- correlation ρ_{12} between the two noises

   XVAR -- variance of the noise n_1; $E[n_1^2]$

   YVAR -- variance of the noise n_2; $E[n_2^2]$

   OUTPUTS: plot of the mean square error as a function of the number of iterations.

2. ANEST -- Main program to implement the LMS algorithm and compute the mean square error; calls subroutines CGAUSS and PLOTIT.

INPUTS: C -- term which determines the rate of convergence.

RO -- correlation ρ_{12} between the two noises.

XVAR -- variance of the noise n_1; $E[n_1^2]$.

YVAR -- variance of the noise n_2; $E[n_2^2]$.

OUTPUTS: plot of the mean square error as a function of the number of iterations.

3. CGAUSS (I1,I2,RO,X,Y) -- generates a pair of zero mean Gaussian random numbers using the Box-Muller equations to transform two uniformly distributed random numbers; calls library uniformly distributed random number generator RANDU.

INPUTS: I1 -- seed number

RO -- desired correlation ρ_{12} between the pair of numbers.

XVAR -- desired variance $E[n_1^2]$ through COMMON statement.

YVAR -- desired variance $E[n_2^2]$ through COMMON statement.

OUTPUTS: X -- zero mean Gaussian random number

Y -- zero mean Gaussian random number

I2 -- seed number.

4. PLOTIT (X,NP,NV,NS) -- plots mean square error as a function of the number of iterations; calls library plotting subroutine TEKPLT.

INPUTS: X -- matrix where each row is a vector of points to be plotted so that up to five curves can be plotted on the same graph; can also be entered as a vector if only one curve is to be plotted.

NP -- number of points within each vector for plotting.

NV -- number of variables or curves to be plotted on the same graph.

NS -- scaling determinant; value of zero allows programmer to scale while value of one provides automatic scaling.

OUTPUTS: plot of vectors entered as a function of the number of points within the vector.

B. Three Input Noise Canceling System

Purpose:

This computer simulation extends the two input system in order to further compare the two algorithms. The equations for the two algorithms are given in Figures 4.4 and 4.5. These were implemented on the PDP-11/45 computer at Duke University. The mean square error was plotted as a function of the number of iterations.

Specifications:

The specifications for the problem were described in Section 1 of Chapter V and are summarized here.

1. The signal is a dc signal with value 10.0.
2. The noise was generated with a method developed by Charles Liu and described in Appendix B. Independent Gaussian random numbers are transformed to correlated ones using this method.
3. The initial weights for the LMS algorithm were set to arbitrary values: $W_2(1) = 5.0$ and $W_3(1) = 3.0$.
4. The mean square error was computed using an iterative estimate:

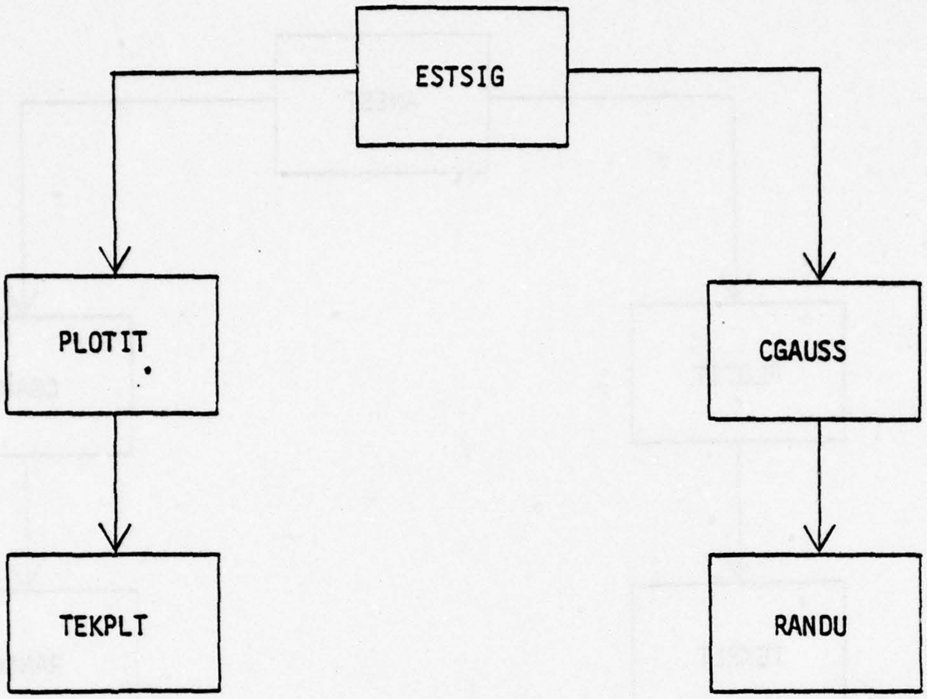


Figure D.1. Computer Program Structure Diagram for the Two Input Noise Canceling System with the Expected Value Algorithm.

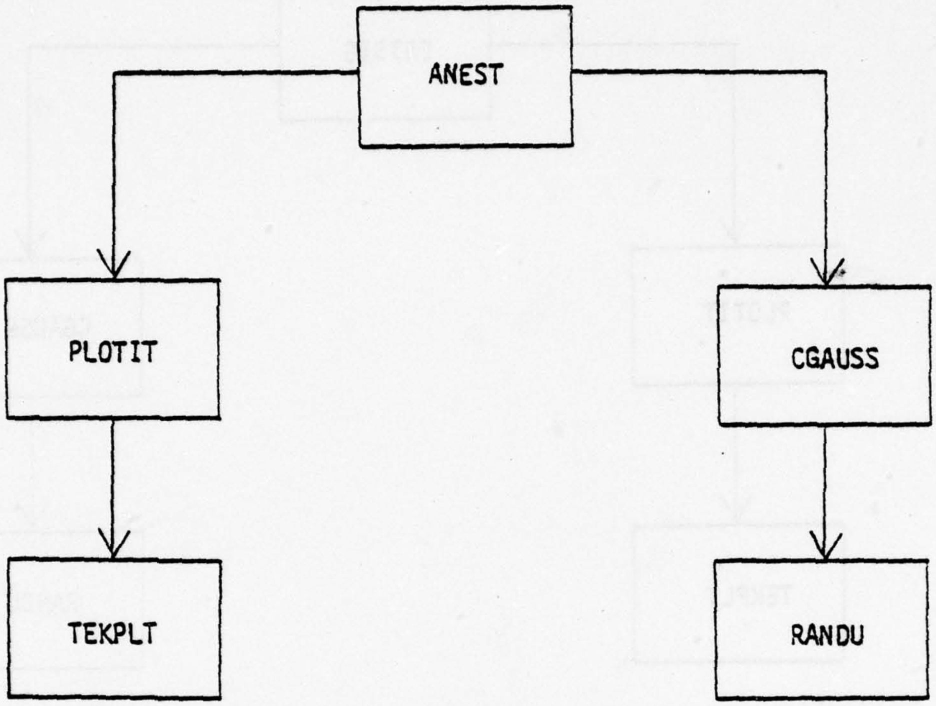


Figure D.2. Computer Program Structure Diagram for the Two Input Noise Canceling System with the LMS Algorithm.

Computer Program Listing

ESTSIG

```
C       ESTSIG -- TO ESTIMATE A SIGNAL
C
        DIMENSION SIG(500),TEMP(500),TMSE(500),TMSE2(500)
        COMMON /A/XVAR,YVAR
        DATA I1,I2/16401,0/
        WRITE (6,90)
        READ (6,91) RO
        WRITE (6,80)
        READ (6,83) XVAR
        WRITE (6,81)
        READ (6,83) YVAR
        NOITER=25
        NS=0
        SIGNAL=10.0
        WRITE (6,94) RO, XVAR, YVAR
        DO 1000 JK=1,500
1000    TMSE(JK)=0.0
        DO 25 I=1,500
25      SIG(I)=SIGNAL
        DO 100 K=1,NOITER
        U1=0.0
        U2=0.0
        DO 101 JKL=1,500
101     TMSE2(JKL)=0.0
        DO 10 I=1,500
        CALL CGAUSS (I1,I2,RO,X,Y)
        X1=SIG(I)+X
        X2=Y
        U1=U1+X1*X2
        U2=U2+X2*X2
        U3=U1/U2
        TEMP(I)=((X1-U3*X2)-SIG(I))**2
        DO 11 J=1,I
11      TMSE2(I)=TMSE2(I) + TEMP(J)
        TMSE2(I)=TMSE2(I)/(FLOAT(I))
10      TMSE(I)=TMSE(I) + TMSE2(I)
100     CONTINUE
        DO 200 IL=1,500
200     TMSE(IL)=TMSE(IL)/(FLOAT(NOITER))
        CALL PLOTIT (TMSE,500,1,NS)
C
80      FORMAT (/1X,'XVAR',/1X,'eeeeeeeeeeeeeeee')
81      FORMAT (/1X,'YVAR',/1X,'eeeeeeeeeeeeeeee')
83      FORMAT (E16.8)
84      FORMAT (/1X,'NOITER',/1X,'eeeee')
86      FORMAT (I5)
90      FORMAT (/1X,'RO',/1X,'eeeeeeeeee')
91      FORMAT (F10.3)
92      FORMAT (/1X,'NS',/1X,'e')
93      FORMAT (I1)
```

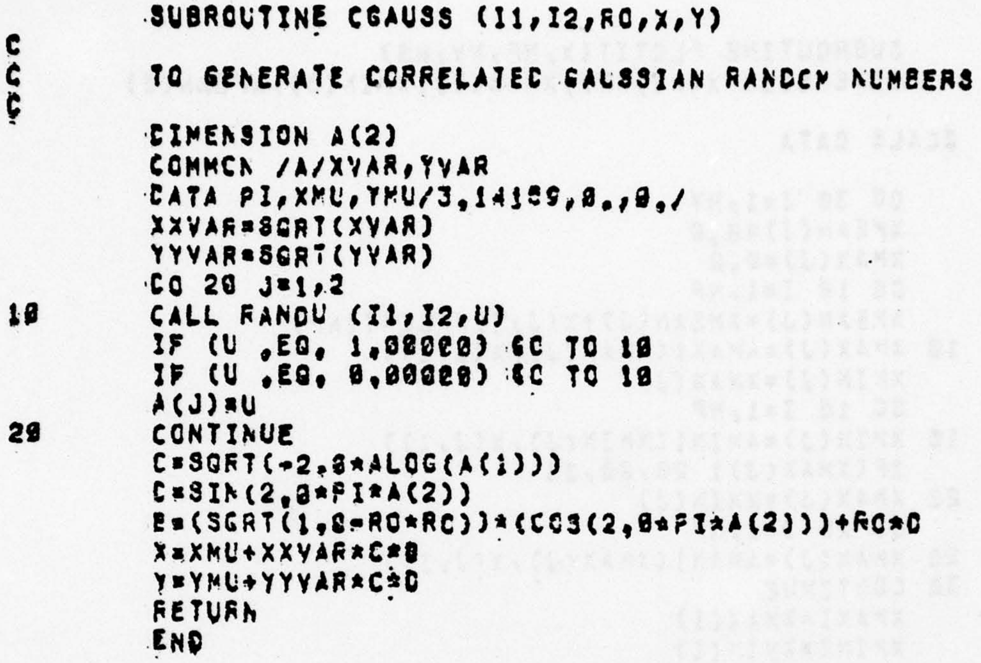
```
94        FORMAT (1X,F10.5,1X,E16.8,1X,E16.8)
96        FORMAT (/1X,'SIGNAL',/1X,'0000000000')
97        FORMAT (F10.1)
3000      CALL EXIT
          END
```

ANEST

```
C
C         ANEST IS A PROGRAM TO ESTIMATE A SIGNAL
C         USING THE LMS ALGORITHM
C
          DIMENSION W(501),SIG(500),TMSE(500),TMSE2(500),TEMP(500)
          COMMON /A/XVAR,YVAR
          DATA I1,I2/16401,0/
          WRITE (6,92)
          READ (6,91) C
          WRITE (6,93)
          READ (6,83) RO
          WRITE (6,85)
          READ (6,91) XVAR
          WRITE (6,86)
          READ (6,91) YVAR
          NOITER=1
          NS=0
          SIGNAL=10.0
          WRITE (6,80) C, RO, XVAR,YVAR
          DO 1000 L=1,500
1000      TMSE(L)=0.0
          DO 25 I=1,500
25        SIG(I)=SIGNAL
          DO 100 K=1,NOITER
          DO 101 JKL=1,500
          TMSE2(JKL)=0.0
101       W(JKL)=0.0
          W(1)=5.0
          DO 60 I=1,500
          CALL CGAUSS (I1,I2,RO,X,Y)
          X1=SIG(I)+X
          X2=Y
          SHAT=X1-W(I)*X2
          W(I+1)=W(I)+2.0*C*X2*(SHAT)
          TEMP(I)=(SHAT-SIG(I))**2
          DO 11 J=1,I
11        TMSE2(I)=TMSE2(I) + TEMP(J)
          TMSE2(I)=TMSE2(I)/(FLOAT(I))
60        TMSE(I)=TMSE(I) + TMSE2(I)
100       CONTINUE
          DO 200 KL=1,500
200       TMSE(KL)=TMSE(KL)/(FLOAT(NOITER))
          CALL PLOTIT (TMSE,500,1,NS)
C
81        FORMAT(I5)
82        FORMAT(/1X,'NOITER',/1X,'======')
80        FORMAT (/1X,E16.8,1X,F10.5,1X,2(E16.8,1X))
91        FORMAT(/1X,'SIGNAL',/1X,'==========')
82        FORMAT(F10.1)
83        FORMAT(F10.5)
```

```
85          FORMAT(/1X,'XVAR',/1X,'eeeeecccceeeeecc')
86          FORMAT(/1X,'YVAR',/1X,'eeeeeceeceeeeece')
90          FORMAT(/1X,'W(1)',/1X,'eeeeeeeeeeeeceece')
91          FORMAT(E16,8)
92          FORMAT(/1X,'C',/1X,'eeeeeeeeceeeees')
93          FORMAT(/1X,'RO',/1X,'eeeeeeeeece')
96          FORMAT(/1X,'NS',/1X,'e')
97          FORMAT(I1)
3000        CALL EXIT
            END
```

CGAUSS



```
      SUBROUTINE CGAUSS (I1,I2,RO,X,Y)
C
C     TO GENERATE CORRELATED GAUSSIAN RANDOM NUMBERS
C
      DIMENSION A(2)
      COMMON /A/XVAR,YVAR
      DATA PI,XMU,YMU/3.14159,0.,0./
      XXVAR=SQRT(XVAR)
      YYVAR=SQRT(YVAR)
      DO 20 J=1,2
10    CALL RANDU (I1,I2,U)
      IF (U .EQ. 1.00000) GO TO 10
      IF (U .EQ. 0.00000) GO TO 10
      A(J)=U
20    CONTINUE
      C=SQRT(-2.0*ALOG(A(1)))
      D=SIN(2.0*PI*A(2))
      B=(SQRT(1.0-RO*RO))*(COS(2.0*PI*A(2)))+RO*D
      X=XMU+XXVAR*C*B
      Y=YMU+YYVAR*C*D
      RETURN
      END
```

PLOTIT

```
      SUBROUTINE PLOTIT(X,NP,NV,NS)
      DIMENSION X(NV,NP),XMAX(5),XMIN(5),XMEAN(5)
C
C  SCALE DATA
C
      DO 30 J=1,NV
      XMEAN(J)=0.0
      XMAX(J)=0.0
      DO 10 I=1,NP
      XMEAN(J)=XMEAN(J)+X(J,I)/FLOAT(NP)
   10 XMAX(J)=AMAX1(XMAX(J),X(J,I))
      XMIN(J)=XMAX(J)
      DO 15 I=1,NP
   15 XMIN(J)=AMIN1(XMIN(J),X(J,I))
      IF(XMAX(J)) 20,20,30
   20 XMAX(J)=XMIN(J)
      DO 25 I=1,NP
   25 XMAX(J)=AMAX1(XMAX(J),X(J,I))
   30 CONTINUE
      XMAXI=XMAX(1)
      XMINI=XMIN(1)
      DO 27 I=1,NV
      XMAXI=AMAX1(XMAXI,XMAX(I))
   27 XMINI=AMIN1(XMINI,XMIN(I))
      IF(NS.EQ.0) GO TO 50
      YINT=(XMAXI-XMINI)/2.+XMINI
   40 IF(ABS(XMAXI-YINT).NE.0.0) GO TO 45
      YS=1.
      GO TO 50
   45 YS=315./ABS(XMAXI-YINT)
   50 XS=800./NP
      IF(NS.NE.3) GO TO 65
      CALL TEKPLT(8,0,0,2,0)
      WRITE(6,2300)
      DO 60 I=1,NV
   60 WRITE(6,2400) I,XMAX(I),XMIN(I),XMEAN(I)
      WRITE(6,2500)
      READ(5,1100) YS
      IF(ABS(YS*XMAXI).GT.32.E+5) YS=32.E+6/XMAXI
      YS=45./YS
      WRITE(6,2600)
      READ(5,1100) YINT
   65 CALL TEKPLT(0,0,0,0,0)
      CALL TEKPLT(2,50,10,850,10)
      CALL TEKPLT(2,50,640,850,640)
      CALL TEKPLT(2,50,10,50,640)
      CALL TEKPLT(2,850,10,850,640)
      CALL TEKPLT(2,50,325,850,325)
      DO 100 J=1,15
      IY=45.*(15-J)+10.
      CALL TEKPLT(2,50,IY,60,IY)
      CALL TEKPLT(2,840,IY,850,IY)
      XP=YINT+(45./YS)*FLOAT(8-J)
      IY=IY-6
      CALL TEKPLT(3,855,IY,0,0)
  100 WRITE(6,2000) XP
      DO 150 J=1,15
      IX=50*J+50
      CALL TEKPLT(2,IX,10,IX,20)
      CALL TEKPLT(2,IX,320,IX,330)
  150 CALL TEKPLT(2,IX,630,IX,640)
      DO 800 J=1,NV
      YY=(X(J,1)-YINT)*YS+325.
        IF(ABS(YY).GE.3.1E+6) YY=3.E+6*YY/ABS(YY)
        IYY=INT(YY)
      CALL TEKPLT(3,5,IYY,5,IYY)
      WRITE(6,2700) J
      DO 800 I=2,NP
      Y1=(X(J,I-1)-YINT)*YS+325.
        IF(ABS(Y1).GE.3.1E+6) Y1=3.E+6*Y1/ABS(Y1)
        IY1=INT(Y1)
      Y2=(X(J,I)-YINT)*YS+325.
        IF(ABS(Y2).GE.3.1E+6) Y2=3.E+6*Y2/ABS(Y2)
        IY2=INT(Y2)
      IX1=(I*XS)+50.
      IX2=(I+1)*XS+50.
  800 CALL TEKPLT(2,IX1,IY1,IX2,IY2)
 1000 FORMAT(1X,I5)
 1100 FORMAT(1X,F10.0)
 2000 FORMAT(1H+,1PE10.3)
 2300 FORMAT(/1X,'VARIABLE',5X,'YMAX',11X,'YMIN',11X,'MEAN'//)
 2400 FORMAT(1X,I5,3(5X,1PE12.3))
 2500 FORMAT(//1X,'ENTER Y-AXIS SCALE'/1X,'(             )')
 2600 FORMAT(//1X,'ENTER X-AXIS POSITION'/1X,'(             )')
 2700 FORMAT(1H+,I2)
 2200 FORMAT(///)
      RETURN
      END
```

$$E[(\hat{s}_j - s_j)^2] = \frac{\sum_{i=1}^{j}(\hat{s}_i - s_i)^2}{j} \qquad \text{(D.1)}$$

5. The mean square error is computed for 500 iterations to produce one ensemble curve. Twenty-five ensembles are averaged point by point and this curve is plotted as a function of the number of iterations.

Program Structure

A separate program is utilized for each algorithm. The programs ESTSI2 and ANEST2 implement the expected value and LMS algorithms, respectively. Each program is divided into three main parts. The first part finds the eigenvalues and eigenvectors of the input correlation matrix so that correlated Gaussian random variables can be generated. The second section generates the noise, computes the weight vector, and calculates the mean square error. The third section averages the ensemble curves and plots the resulting curve. Figures D.3 and D.4 show the structure of the programs.

Program Segment Descriptions

1. ESTSI2 -- Main program to implement the expected value algorithm and compute the mean square error; calls subroutines JACOBI, GAUSS3, and PLOTIT.

   INPUTS: VARNI -- variance of the noise n_1; $E[n_1^2]$.

   OUTPUTS: plot of the mean square error as a function of the number of iterations.

2. ANEST2 -- Main program to implement the LMS algorithm and compute the mean square error; calls subroutines JACOBI, GAUSS3, and PLOTIT.

INPUTS: VARNI -- variance of the noise n_1; $E[n_1^2]$.

CMU -- rate of convergence term, μ.

OUTPUTS: plot of the mean square error as a function of the number of iterations.

3. JACOBI(T, EIGEN, RO12, RO13, RO23)

-- subroutine to compute the eigenvalues and eigenvectors of a symmetric matrix. This program is a variation of the one listed in <u>Applied Numerical Methods</u>. It is used to transform independent Gaussian random numbers to correlated ones.

INPUTS: A(1,2) -- correlation, ρ_{12}, between the noises n_1 and n_2.

A(1,3) -- correlation, ρ_{13}, between the noises n_1 and n_3.

A(2,3) -- correlation, ρ_{23}, between the noises n_2 and n_3.

OUTPUTS: T -- matrix where each column is an eigenvector.

EIGEN -- vector of eigenvalues

RO12 -- correlation, ρ_{12}, between the noises n_1 and n_2.

RO13 -- correlation, ρ_{13}, between the noises n_1 and n_3.

RO23 -- correlation, ρ_{23}, between the noises n_2 and n_3.

list of the elements of the input matrix, A.

4. GAUSS3 (IX,S,AM,F) -- subroutine to generate three independent Gaussian random numbers; calls GAUSS.

INPUTS: IX -- seed number

S -- standard deviation of the variance of each random number; $\sqrt{E[n_1^2]}$.

AM -- desired mean of the generated Gaussian random numbers.

OUTPUTS: F -- vector containing three independent Gaussian random numbers.

5. GAUSS (IX, S, AM, V) -- scientific subroutine package program to generate one independent Gaussian random number; calls library subroutine RANDU.

INPUTS: IX -- seed number

S -- standard deviation of the desired random number.

AM -- desired mean of the random number.

OUTPUTS: V -- Gaussian random number generated.

6. PLOTIT (X, NP, NV, NS) -- plots mean squares error as a function of the number of iterations; calls library plotting subroutine TEKPLT.

INPUTS: X -- matrix where each row is a vector of points to be plotted so that up to five curves can be plotted on the same graph; can also be entered as a vector if only one curve is to be plotted.

NP -- number of points within each vector to be plotted.

NV -- number of variables or curves to be plotted on the same graph.

NS -- scaling determinant; value of zero allows programmer to scale while value of one provides automatic scaling.

OUTPUTS: plot of vectors entered as function of the number of points within the vector.

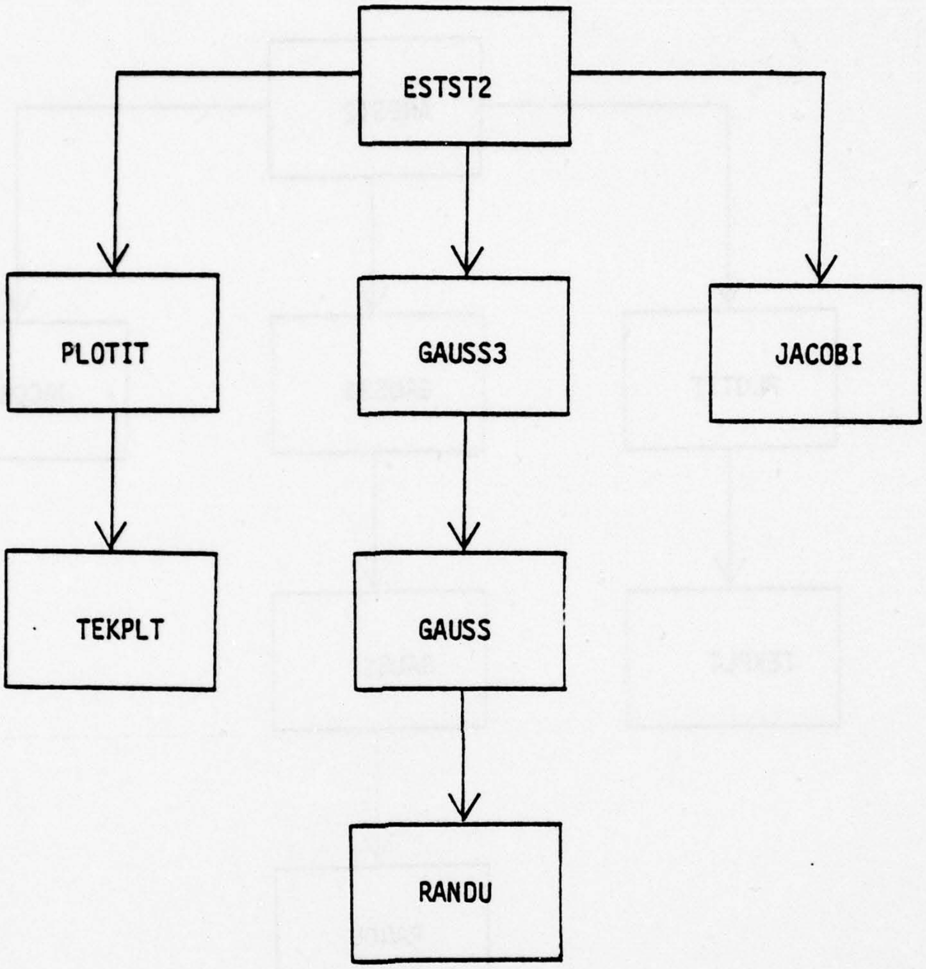


Figure D.3. Computer Program Structure Diagram for the Three Input Noise Canceling System with the Expected Value Algorithm.

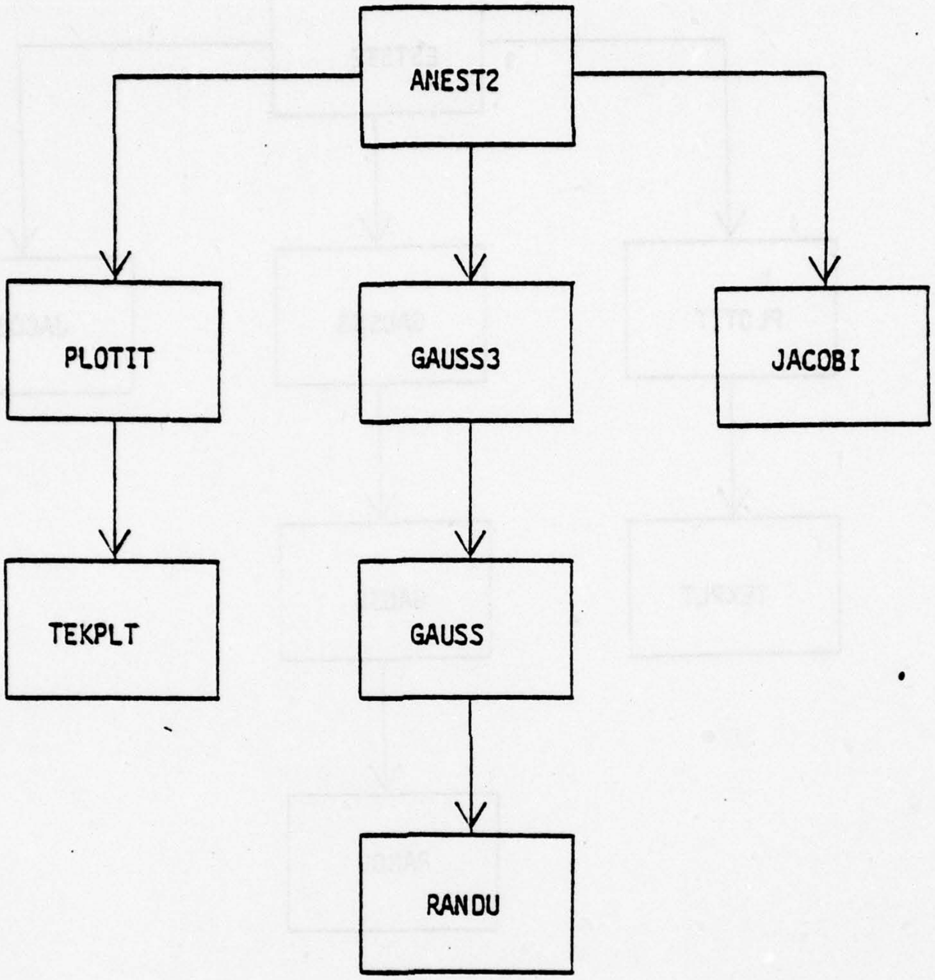


Figure D.4. Computer Program Structure Diagram for the Three Input Noise Canceling System with the LMS Algorithm.

Computer Program Listing

ESTSI2

```
C         ESTSI2 -- TO ESTIMATE A SIGNAL USING THE EXPECTED
C                   VALUE ALGCRITHM ANC TWO NOISE ONLY
C                   INPUTS
C
          CIMENSION SIG(500),TEMP(500),TMSE(500),TMSE2(500)
          CIMENSION CIAG(3),Y(3),VECS(3,3),VALS(3),F(3),EDEN(3,3)
          CIMENSION EIGEN(3),T(3,3),TRANS(3,3)
          COUBLE PRECISICN VECS,VALS,DRC12,DRO13,CRO23
          CATA IX,AM/16481,0.2/
          WRITE (6,80)
          READ (6,83) VARN1
          WRITE (6,81)
          S=SCRT(VARN1)
          NOITER=25
          CC 1000 JK=1,500
          SIG(JK)=10.0
1000      TMSE(JK)=0.0
          CO 10 I=1,3
          CO 10 J=1,3
10        EDEN(I,J)=0.0
          CALL JACOBI (VECS,VALS,DRO12,CRC13,DRC23)
          DC 22 I=1,3
          EIGEN(I)=SNGL(VALS(I))
          CO 20 J=1,3
20        T(I,J)=SNGL(VECS(I,J))
          RC12=SNGL(CRO12)
          RO13=SNGL(CRC13)
          RO23=SNGL(CRC23)
          CO 21 I=1,3
21        CIAG(I)=SSRT(EIGEN(I))
          CC 22 I=1,3
22        EDEN(I,I)=CIAG(I)
          CO 15 I=1,3
          CO 15 J=1,3
15        TRANS(I,J)=T(I,1)*ECEN(1,J)+T(I,2)*ECEN(2,J)+
     *    T(I,3)*EDEN(3,J)
          CO 100 K=1,NCITER
          A12=0.0
          A13=0.0
          A22=0.001
          A23=0.0
          A33=0.0
          CO 101 JKL=1,500
101       TMSE2(JKL)=0.0
          CC 200 I=1,500
          CALL GAUSS3 (IX,S,AM,F)
          DO 17 N=1,3
17        Y(N)=TRANS(N,1)*F(1)+TRANS(N,2)*F(2)+TRANS(N,3)*F(3)
          X1=SIG(I)+Y(1)
          X2=Y(2)
```

```
      X3=Y(3)
      A12=A12+X1*X2
      A13=A13+X1*X3
      A22=A22+X2*X2
      A33=A33+X3*X3
      A23=A23+X2*X3
      WDEN=A22*A33-A23*A23
      W2=(A12*A33-A13*A23)/WDEN
      W3=(A13*A22-A12*A23)/WDEN
      TEMP(I)=((X1-W2*X2-W3*X3)-SIG(I))**2
      DO 11 J=1,I
11    TMSE2(I)=TMSE2(I) + TEMP(J)
      TMSE2(I)=TMSE2(I)/(FLOAT(I))
200   TMSE(I)=TMSE(I) + TMSE2(I)
100   CONTINUE
      DO 201 IL=1,500
201   TMSE(IL)=TMSE(IL)/(FLOAT(NCITER))
      CALL PLOTIT (TMSE,500,1,0)
C
80    FORMAT (/1X,'VARN1',/1X,'-----------------')
81    FORMAT(/6X,'A(1,2)','A(1,3)','A(2,3)',/6X,'-------',
     1'-------','-------')
83    FORMAT (E16,8)
90    FORMAT (F3,1)
3000  CALL EXIT
      END
```

ANEST2

```
C        ANEST2 -- TO ESTIMATE A SIGNAL USING THE LMS
C                  ALGORITHM AND TWO NOISE ONLY INPUTS
C
C
      DIMENSION SIG(500),TEMP(500),TMSE(500),TMSE2(500)
      DIMENSION DIAG(3),Y(3),VECS(3,3),VALS(3),F(3),EDEN(3,3)
      DIMENSION EIGEN(3),T(3,3),TRANS(3,3),W2(501),W3(501)
      DOUBLE PRECISION VECS,VALS,DRO12,DRO13,DRO23
      DATA IX,AM/16401,0.0/
      WRITE (6,80)
      READ (6,83) VARN1
      WRITE (6,84)
      READ (6,83) CMU
      WRITE (6,81)
      S=SQRT(VARN1)
      NOITER=25
      DO 1000 JK=1,500
      SIG(JK)=10.0
 1000 TMSE(JK)=0.0
      DO 10 I=1,3
      DO 10 J=1,3
   10 EDEN(I,J)=0.0
      CALL JACOBI (VECS,VALS,DRO12,DRO13,DRO23)
      DO 20 I=1,3
      EIGEN(I)=SNGL(VALS(I))
      DO 20 J=1,3
   20 T(I,J)=SNGL(VECS(I,J))
      RO12=SNGL(DRO12)
      RO13=SNGL(DRO13)
      RO23=SNGL(DRO23)
      DO 21 I=1,3
   21 DIAG(I)=SQRT(EIGEN(I))
      DO 22 I=1,3
   22 EDEN(I,I)=DIAG(I)
      DO 15 I=1,3
      DO 15 J=1,3
   15 TRANS(I,J)=T(I,1)*EDEN(1,J)+T(I,2)*EDEN(2,J)+
     * T(I,3)*EDEN(3,J)
      DO 100 K=1,NOITER
      DO 101 JKL=1,500
      W2(JKL)=0.0
      W3(JKL)=0.0
  101 TMSE2(JKL)=0.0
      W2(1)=5.0
      W3(1)=3.0
      DO 200 I=1,500
      CALL GAUSS3 (IX,S,AM,F)
      DO 17 N=1,3
   17 Y(N)=TRANS(N,1)*F(1)+TRANS(N,2)*F(2)+TRANS(N,3)*F(3)
      X1=SIG(I)+Y(1)
```

THIS PAGE IS BEST QUALITY PRACTICABLE
FROM COPY FURNISHED TO DDC

```
          X2=Y(2)
          X3=Y(3)
          SHAT=X1-W2(I)*X2-W3(I)*X3
          W2(I+1)=W2(I)+2,0*CMU*SHAT*X2
          W3(I+1)=W3(I)+2,0*CMU*SHAT*X3
          TEMP(I)=(SHAT-SIG(I))**2
          DO 11 J=1,I
11        TMSE2(I)=TMSE2(I) + TEMP(J)
          TMSE2(I)=TMSE2(I)/(FLOAT(I))
200       TMSE(I)=TMSE(I) + TMSE2(I)
130       CONTINUE
          DO 201 IL=1,500
201       TMSE(IL)=TMSE(IL)/(FLOAT(NOITER))
          CALL PLOTIT (TMSE,500,1,0)
C
80        FORMAT (/1X,'VARN1',/1X,'ØØØØØØØØØØØØØØØØ')
81        FORMAT(/6X,'A(1,2)','A(1,3)','A(2,3)',/6X,'ØØØØØØ',
        1'ØØØØØØØ','ØØØØØØØ')
83        FORMAT (E16,8)
84        FORMAT (/1X,'CMU',/1X,'ØØØØØØØØØØØØØØØØØ')
90        FORMAT (F3,1)
3000      CALL EXIT
          END
```

JACOBI

```
      SUBROUTINE JACOBI (T,EIGEN,RC12,RC13,RO23)
C     EIGENVALUES AND EIGENVECTCRS BY THE JACCBI METHOD
C     APPLIED NUBERICAL METHCDS, EXAMPLE 4,4
C
      IMPLICIT REAL*8 (A-H),(O-Z)
      CIMENSION A(3,3),T(3,3),AIK(3),EIGEN(3)
      CATA N,ITMAX/3,10/
      DATA EPS1,EPS2,EPS3/1,00D-10,1,00C-10,1,00C-5/
C
C     READ PARAMETERS AND ESTABLISH STARTING MATRIX A
      CO 2 I=1,3
      CO 2 J=1,3
      T(I,J)=0,0
2     A(I,J)=0,0
      READ (6,102) A(1,2),A(1,3),A(2,3)
      A(2,1)=A(1,2)
      A(3,1)=A(1,3)
      A(3,2)=A(2,3)
      A(1,1)=1,00
      A(2,2)=1,00
      A(3,3)=1,00
      RO12=A(1,2)
      RC13=A(!,3)
      RO23=A(2,3)
      WRITE (6,101) ((A(I,J),J=1,N),I=1,N)
      NM1=N-1
C
C     SET UP INITIAL MATRIX T, COMPUTE SIGMA1 ANC S
      SIGMA1=0,0
      CFFDSG=0,0
      CO 5 I=1,N
      SIGMA1=SIGMA1+A(I,I)**2
      T(I,I)=1,0
      IP1=I+1
      IF (I ,GE, N) GO TO 6
      CO 5 J=IP1,N
5     CFFDSG=OFFCSG+A(I,J)**2
6     S=2,0*OFFDSG+SIGMA1
C
C     BEGIN JACOBI ITERATICN
      DO 26 ITER=1,ITMAX
      CO 20 I=1,NM1
      IP1=I+1
      CO 20 J=IP1,N
      G=DABS(A(I,I)-A(J,J))
C
C     COMPUTE SINE AND COSINE OF RCTATION ANGLE
      IF (G ,LE, EPS1) GO TC 9
      IF (CABS(A(I,J)) ,LE, EPS2) GC TO 20
      F=2,0*A(I,J)*G/(A(I,I)-A(J,J))
      SPQ=DSQRT(P*P+Q*Q)
      CSA=DSQRT((1.0+Q/SPQ)/2.0)
      SNA=P/(2.0*CSA*SPQ)
      GO TO 10
9     CSA=1.0/DSQRT(2.0D0)
      SNA=CSA
10    CONTINUE
C
C     UPDATE COLUMSN I AND J OF T - EQUIVALENT TO
C     MULTIPLICATION BY THE ANNIHILATION MATRIX
      DO 11 K=1,N
      HOLDKI=T(K,I)
      T(K,I)=HOLDKI*CSA+T(K,J)*SNA
11    T(K,J)=HOLDKI*SNA-T(K,J)*CSA
C
C     COMPUTE NEW ELEMENTS OF A IN ROWS I AND J
      DO 16 K=I,N
      IF (K .GT. J) GO TO 15
      AIK(K)=A(I,K)
      A(I,K)=CSA*AIK(K)+SNA*A(K,J)
      IF (K .NE. J) GO TO 14
        A(J,K)=SNA*AIK(K)-CSA*A(J,K)
14    GO TO 16
15    HOLDIK=A(I,K)
      A(I,K)=CSA*HOLDIK+SNA*A(J,K)
      A(J,K)=SNA*HOLDIK-CSA*A(J,K)
16    CONTINUE
C
C     COMPUTE NEW ELEMENTS OF A IN COLUMNS I AND J
      AIK(J)=SNA*AIK(I)-CSA*AIK(J)
C
C     WHEN K IS LARGER THAN I
      DO 19 K=1,J
      IF (K .LE. I) GO TO 18
      A(K,J)=SNA*AIK(K)-CSA*A(K,J)
      GO TO 19
18    HOLDKI=A(K,I)
      A(K,I)=CSA*HOLDKI+SNA*A(K,J)
      A(K,J)=SNA*HOLDKI-CSA*A(K,J)
19    CONTINUE
20    A(I,J)=0.0
C
C     FIND SIGMA2 FOR TRANSFORMED A AND TEST FOR CONVERGENCE
      SIGMA2=0.0
      DO 21 I=1,N
      EIGEN(I)=A(I,I)
21    SIGMA2=SIGMA2+EIGEN(I)**2
      IF (1.0-SIGMA1/SIGMA2 .GE. EPS3) GO TO 26
        GO TO 300
26    SIGMA1=SIGMA2
C
C     IF ITER EXCEEDS ITMAX, NO CONVERGENCE
      WRITE (6,203) ITER,S,SIGMA1,SIGMA2
      GO TO 1
C
C     FORMATS FOR INPUT AND OUPUT STATEMENTS
101   FORMAT (10X,6F10.4)
102   FORMAT (5X,3F6.4)
203   FORMAT(/5X,' NO CONVERGENCE, WITH',//7X,
     *'ITER    = ',I5,5X,' S         = ',F10.5,5X,
     *' SIGMA1 = ',F10.5,5X,' SIGMA2 = ',F10.5,//5X,
     *' THE CURRENT A MATRIX IS',/1X)
C
300   RETURN
      END
```

GAUSS3

```
      SUBROUTINE GAUSS3 (IX,S,AM,F)
C     TO GENERATE 3 INDEPLENDENT GAUSSIAN RANDOM NUMBERS
C
      DIMENSION F(3)
      DO 10 I=1,3
      CALL GAUSS (IX,S,AM,V)
10    F(I)=V
      RETURN
      END
```

GAUSS

```
        SUBROUTINE GAUSS(IX,S,AM,V)
        A=Ø.Ø
        DO 5Ø I=1,12
        CALL RANDU (IX,IY,Y)
        IX=IY
5Ø      A=A+Y
        V=(A-6.Ø)*S+AM
        RETURN
        END
```

LIST OF REFERENCES

REFERENCES

Brauer, Fred and Nohel, John A., 1969. The Qualitative Theory of Ordinary Differential Equations. W. A. Benjamin, Inc., Menlo Park, California.

Bullier, Jean, 1976-7. Private communications.

Carnahan, Brice, Luther, H. A., and Wilkes, James O., 1969. "Jacobi's Method for Symmetric Matrices," Applied Numerical Methods. John Wiley and Sons, Inc., New York, New York.

Clark, Gregory A., September 27, 1976. "Cancel 60 Hz and Other Noise," Electronic Design, Vol. 24, No. 20, pp. 74-79.

Hohn, Franz E., 1972. Introduction to Linear Algebra. The MacMillan Company, New York, New York.

Huffman, Steven D., 1976. Private communications.

Jansson, Birger, 1964. "Generation of Random Bivariate Normal Deviates and Computation of Related Integrals," Nordisk Tidfkrift for Informations Behandling, Vol. 4, pp. 205-212.

Kailath, Thomas, March 1974. "A View of Three Decades of Linear Filtering Theory," IEEE Transactions on Information Theory, Vol. IT-20, No. 2, pp. 146-181.

Lathi, B. P., 1968. An Introduction to Random Signals and Communication Theory. International Textbook Company, Scranton, Pennsylvania.

Lee, Sang C., 1977. Private communications.

Liu, Charles S., 1976. "Generating Random Numbers Having Correlated Covariance." Technical Note, Department of Electrical Engineering, University of Michigan-Dearborn, Dearborn, Michigan.

Liu, Charles S., 1977. Private communications.

Papoulis, A. N., 1965. Probability, Random Variables, and Stochastic Processes. McGraw-Hill Book Co., New York, New York.

Widrow, B., Glover, Jr., J. R., McCool, J. M., Kaunitz, J., Williams, C. S., Hearn, R. H., Zeidler, J. R., Dong, Jr., E., and Goodlin, R. C., 1975. "Adaptive Noise Cancelling: Principles and Applications," Proceedings of the IEEE, Vol. 63, No. 12, pp. 1692-1716.

Widrow, B., McCool, J. M., Larimore, M. G., and Johnson, Jr., 1976. "Stationary and Nonstationary Learning Characteristics of the LMS Adaptive Filter," Proceedings of the IEEE, Vol. 64, No. 8, pp. 1151-1162.

Unclassified

Security Classification

DOCUMENT CONTROL DATA - R & D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY (Corporate author)

Department of Electrical Engineering ✓
Duke University
Durham, North Carolina 27706

2a. REPORT SECURITY CLASSIFICATION

Unclassified

2b. GROUP

3. REPORT TITLE

6 A Comparison of Two Noise Canceling System Algorithms,

4. DESCRIPTIVE NOTES (Type of report and inclusive dates)

Technical report

9 Masters thesis,

5. AUTHOR(S) (First name, middle initial, last name)

10 Rachel Patricia/Messina

12 132p.

6. REPORT DATE

11 Dec~~ember~~ 1977

7a. TOTAL NO. OF PAGES

118

7b. NO. OF REFS

15

8a. CONTRACT OR GRANT NO.

15 N00014-75-C-0191 ✓

b. PROJECT NO.

c.

d.

9a. ORIGINATOR'S REPORT NUMBER(S)

Adaptive Signal Detection Laboratory

9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report)

14 TR-14 ✓

10. DISTRIBUTION STATEMENT

Distribution of this document is unlimited.

11. SUPPLEMENTARY NOTES

12. SPONSORING MILITARY ACTIVITY

Office of Naval Research
Navy Department (Code 222)

13. ABSTRACT

The purpose of this study is to compare two noise canceling system algorithms when the system has a small number of inputs. The expected value and LMS (least mean square) algorithms attempt through different methods to determine the optimum solution given by the Wiener-Hopf weight vector. The expected value algorithm estimates the necessary expected values, substitutes them into the Wiener-Hopf equation, and performs the necessary matrix inversion. The LMS algorithm uses gradient techniques and the method of steepest descent to avoid the complexity of the matrix inversion. The algorithms are compared for two and three input systems in terms of the rate of convergence of the mean square error and the minimum error after five hundred iterations.

A computer simulation of a dc signal in Gaussian noise provided the comparison between the two algorithms. The LMS algorithm presented problems in implementation because of two parameter values that had to be chosen. Although a value for the initial weights had to be determined, this value was not as important as was the value of μ, the term that controls the rate of convergence. This term must be within a specific range in order to guarantee convergence of the algorithm.

In conclusion, the expected value algorithm was judged to be the better of the two. The tradeoff in terms of performance for simplicity in the LMS algorithm was great especially when the signal-to-noise ratio or the correlation between the noises was small. If a "correct" value of μ is chosen, the results may be similar, but if not, the results can be extremely different from those of the expected value algorithm. Thus for a small number of inputs, where the matrix to be inverted is of small dimension, it is better in terms of performance to use the approximation to the optimum solution given by the expected value algorithm.

405708

Unclassified

Security Classification

B

14. KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
Signal detection						
Signal processing						
Estimation						
Bayesian						
Likelihood ratio test						
Gaussian						
Noise Canceling System						
Wiener-Hopf equation						
Information theory						
LMS algorithm						

UNCLASSIFIED

DISTRIBUTION LIST

Office of Naval Research
800 N. Quincy Street
Arlington, Virginia 22217
Attn: Code 222 2
102 OS 1
480 1

Director
Naval Research Laboratory
Technical Information Division
4555 Overlook Avenue S.W.
Washington, D.C. 20375 6

Director
Office of Naval Research Branch Office
1030 East Green Street
Pasadena, California 91106 1

Office of Naval Research
San Francisco Area Office
760 Market Street Room 447
San Francisco, California 94102 1

Director
Office of Naval Research Branch Office
495 Summer Street
Boston, Massachusetts 02210 1

Office of Naval Research
New York Area Office
207 West 24th Street
New York, New York 10011 1

Commanding Officer
Office of Naval Research Branch Office
Box 39
FPO New York 09510 1

Director
Office of Naval Research Branch Office
536 South Clark Street
Chicago, Illinois 60605 1

Commander
Naval Surface Weapons Center
Acoustics Division
Silver Spring, Maryland 20910
Attn: Dr. Zaka Slawsky 1

Officer in Charge
Naval Ship Research & Development Center
Annapolis Laboratory
Annapolis, Maryland 21402 1

Commander
Naval Sea Systems Command
Department of the Navy
Washington, D.C. 20362
Attn: SEA 037 1
Carey Smith, 06H1 1

Commanding Officer
Fleet Numerical Weather Central
Monterey, California 93940 1

Defense Documentation Center
Cameron Station
Alexandria, Virginia 22314 12

Director of Navy Laboratories
Chief of Naval Material
2211 Jefferson Davis Highway
Crystal Plaza #5
Arlington, Virginia 20360
AttnDr.James Probus NAVMAT 03L 1

Commander
Naval Electronic Systems Command
2511 Jefferson Davis Highway
Arlington, Virginia 20360
Attn: CDR A. R. Miller NAVELEX 320 1

Commander
Naval Ship Research & Development Center
Department of the Navy
Bethesda, Maryland 20084
Attn: Mr. Craig Olson 1
Unclassified Library 1

Chief of Naval Operations
Room 4D518, Pentagon
Washington, D.C. 20350
Attn: CAPT A. H. Gilmore 1

Commander
Naval Ocean Systems Center
Department of the Navy
San Diego, California 92132
Attn: Dr. Dan Andrews 1
Mr. Henry Aurand 1
Dr. Dean Hanna 1

Superintendent
Naval Research Laboratory
Underwater Sound Reference Division
P.O. Box 8337
Orlando, Florida 32806 1

Commanding Officer
Naval Underwater Systems Center
New London Laboratory
New London, Connecticut 06320
Attn: Dr. A. Nuttall 1
Mr. A. Ellinthorpe 1
Dr. D. M. Viccione 1

Commander
Naval Air Development Center
Department of the Navy
Warminster, Pennsylvania 18974
Attn: Unclassified Library 1

Superintendent
Naval Postgraduate School
Monterey, California 93940
Attn: Unclassified Library 1

Commanding Officer
Naval Coastal Systems Laboratory
Panama City, Florida 32401
Attn: Unclassified Library 1

Commanding Officer
Naval Underwater Systems Center
Newport Laboratory
Newport, Rhode Island 02840
Attn: Unclassified Library 1

Superintendent
U.S. Naval Academy
Annapolis, Maryland 21402
Attn: Library 1

Commanding Officer
Naval Intelligence Support Center
4301 Suitland Road
Suitland, Maryland 20390
Attn: Dr. Johann Martinek 1
Mr. E. Bissett 1

Commander
Naval Sea Systems Command
Washington, D.C. 20362
Attn: Unclassified Library, SEA 03E 1

Office of the Assistant Secretary of the Navy
for Research, Engineering & Systems
Room 4E732, Pentagon
Washington, D.C. 20350
Attn: Mr. Gerald Cann 1

Special Assistant for ASW
Office of the Assitant Secretary of the Navy
for Research, Engineering & Systems
Washington, D.C. 20350
Attn: Dr. D. Hyde 1

Dr. Melvin J. Jacobson
Rennselaer Polytechnic Institute
Troy, New York 12181 1

Dr. Charles Stutt
General Electric Company
P.O. Box 1088
Schenectady, New York 12301 1

Dr. Alan Winder
MSB Systems, Inc.
25 Sylvan Road South
West Point, Connecticut 06880 1

Dr. T. G. Birdsall
Cooley Electronics Laboratory
University of Michigan
Ann Arbor, Michigan 48105 1

Dr. Harry DeFerrari
University of Miami
Rosentiel School of Marine & Atmospheric Sciences
4600 Rickenbacker Causeway
Miami, Florida 33149 1

Mr. Robert Cunningham
Bendix Electronics Center
15825 Roxford Street
Sylmar, California 91342 1

Dr. Stephen Wolff
Johns Hopkins University
Baltimore, Maryland 21218 1

Dr. M. A. Basin
S.D.P., Inc
15250 Ventura Boulevard
Suite 518
Sherman Oaks, California 91403 1

Dr. Walter Duing
University of Miami
Rosentiel School of Marine & Atmospheric Sciences
4600 Rickenbacker Causeway
Miami, Florida 33149 1

Dr. David Middleton
127 East 91st Street
New York, New York 10028 1

Dr. Donald W. Tufts
University of Rhode Island
Kingston, Rhode Island 02881 1

Dr. Loren Nolte
Duke University
Department of Electrical Engineering
Durham, North Carolina 27706 1

Mr. S. W. Autrey
Hughes Aircraft Company
P.O. Box 3310
Fullerton, California 92634 1

Dr. Thomas W. Ellis
Texas Instruments, Inc
13500 North Central Expressway
Dallas, Texas 75231 1

Dr. Terry Ewart
Applied Physics Laboratory
University of Washington
1013 Northeast Fortieth Street
Seattle, Washington 98195 1

Institute for Acoustical Research
Miami Division of the Palisades Geophysical Institute
615 S.W. 2nd Avenue
Miami, Florida 33130
Attn: Mr. M. Kronengold 2
Dr. J. Clark
Dr. W. Jobst
Dr. S. Adams

Mr. Carl Hartdegen
Palisades Geophysical Institute
Sofar Station
FPO New York 09560 1

Mr. Charles Loda
Institute for Defense Analyses
400 Army-Navy Drive
Arlington, Virginia 22202 1

Mr. Beaumont Buck
Polar Research Laboratory
123 Santa Barbara Avenue
Santa Barbara, California 93101 1

Dr. M. Weinstein
Underwater Systems, Inc
8121 Georgia Avenue
Silver Spring, Maryland 20910 1

Dr. Thomas G. Kincaid
General Electric Company
P.O. Box 1088
Schenectady, New York 12301 1

Applied Research Laboratories
University of Texas at Austin
P.O. Box 8029
10000 FM Road
Austin, Texas 78712
Attn: Dr. Lloyd Hampton 2
Dr. Charles Wood

Woods Hole Oceanographic Institution
Woods Hole, Massachusetts 02543
Attn: Dr. Paul McElroy 1
Dr. R. Porter
Dr. R. Spindel

Dr. John Bouyoucos
Hydroacoustics, Inc
321 Northland Avenue
P.O. Box 3818
Rochester, New York 14610 1

Systems Control, Inc
260 Sheridan Avenue
Palo Alto, California 94306
Attn: Mr. Robert Baron 1

Atlantic Oceanographic & Meteorological Laboratories
15 Rickenbacker Causeway
Miami, Florida 33149
Attn: Dr. John Proni 1

Dr. C. N. K. Mooers
University of Delaware
Newark, Delaware 19711 1

Westinghouse Electric Corporation
Advanced Development Program
Marketing Department MS 227
P.O. Box 746
Baltimore, Maryland 21203
Attn: F. J. Frissyn 1

Oak Ridge National Laboratory
Union Carbide Corporation
Nuclear Division
P.O. Box X
Oak Ridge, Tennessee 37830 1

Professor Neil Bershad
University of California, Irvine
School of Engineering
Irvine, California 92664 1